

Chapitre 2 : Architecture de base d'un ordinateur

❑ Catégories des ordinateurs

- **les gros systèmes** : les mainframes, les supercalculateurs
(gros calculs : recherche, industrie, météo...)
- **les ordinateurs en entreprise** : serveurs applicatifs métier, serveurs intranet ou Internet, etc.
- **les ordinateurs de systèmes embarqués** : voitures, véhicules ou machines industrielles,...



❑ Catégories des ordinateurs

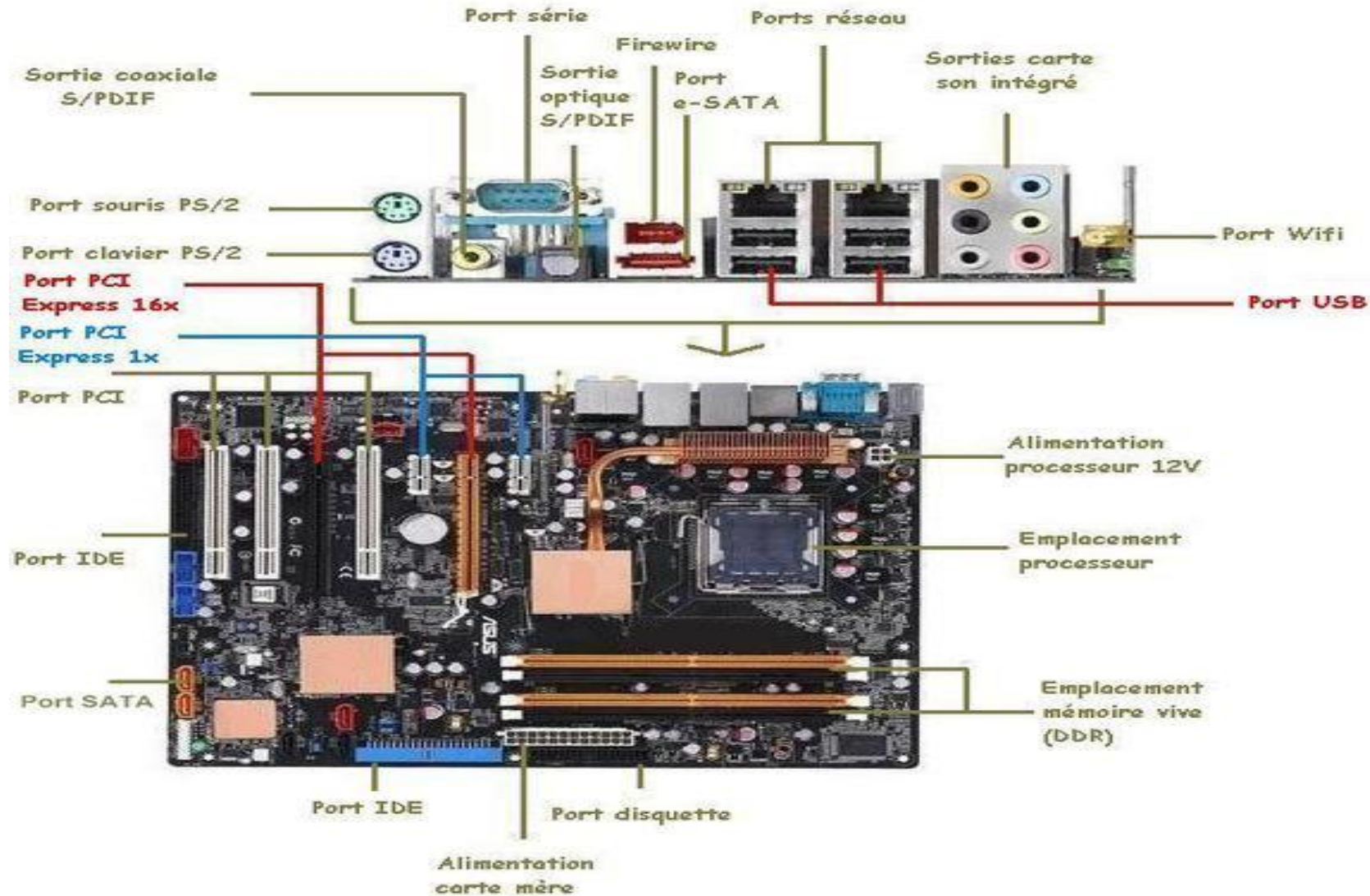
- **les ordinateurs individuels et familiaux** : appelés micro-ordinateurs fixes (bureautique, Internet, jeux...). Le premier PC est né chez IBM en 1980.
- **les micro-ordinateurs portables** : facilement transportables et autonomes
- **les ordinateurs de poche** : PDA(Personal Digital Assistant) ou assistants personnels



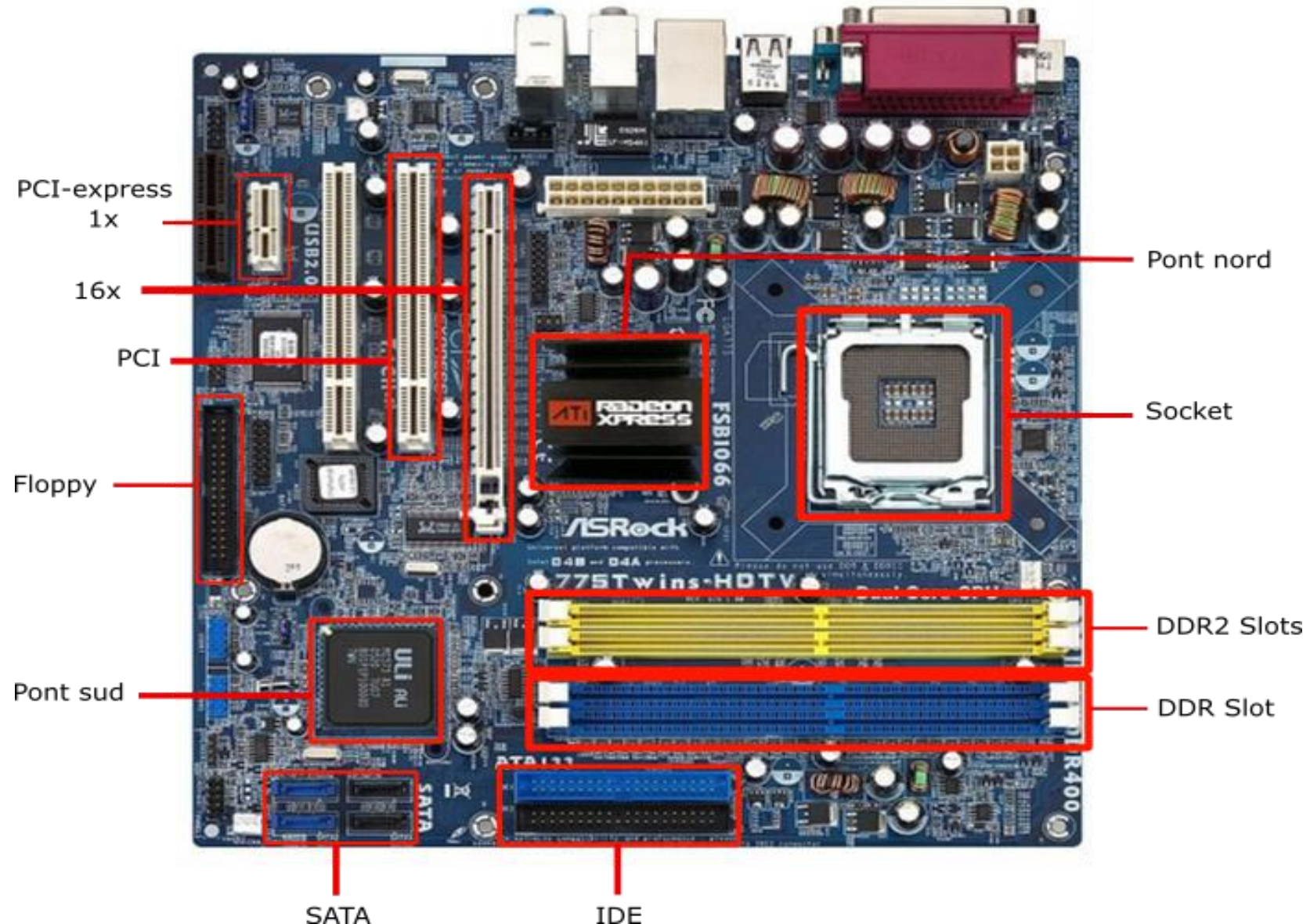
❑ Composants d'un ordinateur personnel

- Unité centrale
 - Carte mère
 - Processeur
 - Mémoire
 - Carte graphique
 - Périphérique interne de stockage
- Moniteur
- Périphériques
- clavier, souris, etc...

❑ Carte mère



❑ Carte mère



❑ Carte mère

- La carte mère est un circuit imprimé qui permet de mettre en contact physique les différents composants et périphériques.
- C'est un socle permettant la **connexion** de l'ensemble des éléments essentiels de l'ordinateur
- Caractéristiques:
 - **le facteur d'encombrement** : la géométrie, les dimensions, l'agencement et les caractéristiques électriques de la carte-mère
 - **le chipset** (jeu de composants ou jeu de circuits) : circuit électronique chargé de coordonner les échanges de données entre les divers composants
 - **le type de support de processeur**
 - **les connecteurs d'entrée-sortie**

❑ Carte mère: Les composants intégrés

- La carte mère contient un certain nombre d'**éléments embarqués**, i.e. intégrés sur son circuit imprimé :
 - **Le chipset** (ou pont) circuit qui contrôle la majorité des ressources
 - L'horloge et la pile du CMOS
 - Le **BIOS**
 - Le bus système et les bus d'extension
- Les cartes mères récentes embarquent généralement un certain nombre de périphériques multimédia et réseau pouvant être désactivés :
 - carte réseau intégrée
 - carte graphique intégrée
 - carte son intégrée
 - contrôleurs de disques durs évolués

❑ Chipset

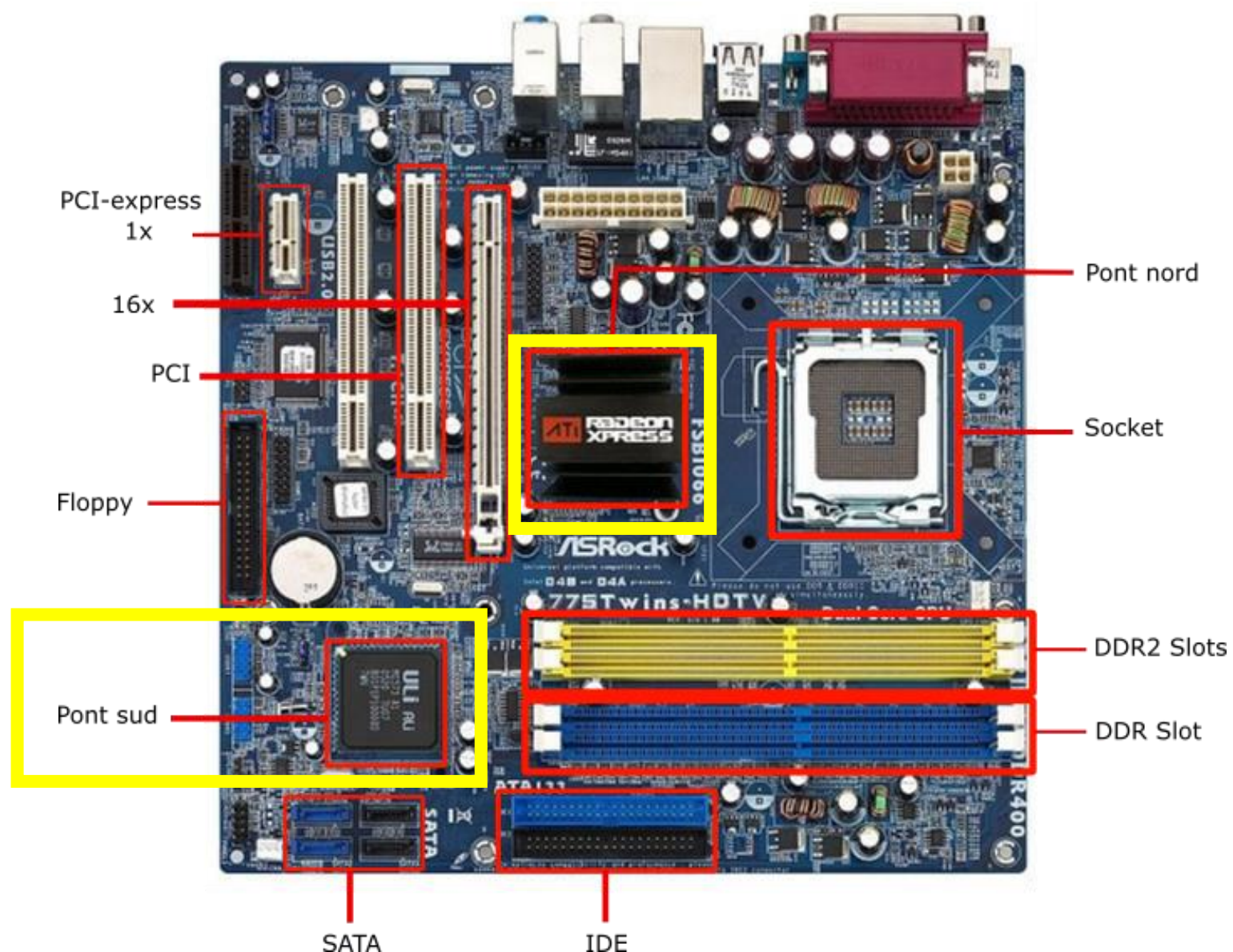
- Un **jeu de composants électroniques** permettant de gérer les **flux de données** numériques entre le processeur, la mémoire et les périphériques.
- Se trouve dans les appareils électroniques de type micro-ordinateur, console de jeux vidéo, téléphone mobile, appareil photographique numérique, GPS, etc.
- **La carte mère et le chipset déterminent :**
 - la vitesse des différents bus
 - le type de processeur qui peut être utilisé et la gamme de fréquences
 - le type de mémoire qui peut être utilisé, ainsi que la taille maximale de la mémoire
 - les périphériques qui sont susceptibles d'être connectés

❑ Chipset

- **Pont Northbridge:** il gère des communications entre le microprocesseur et les périphériques rapides tels que:
 - la mémoire vive (RAM)
 - les bus AGP (Accelerated Graphics Port)
 - les bus PCI (Peripheral Component Interconnect) Express
 - le pont sud.

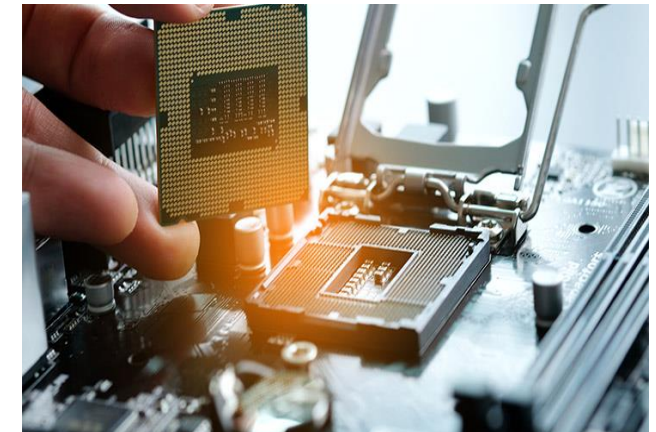
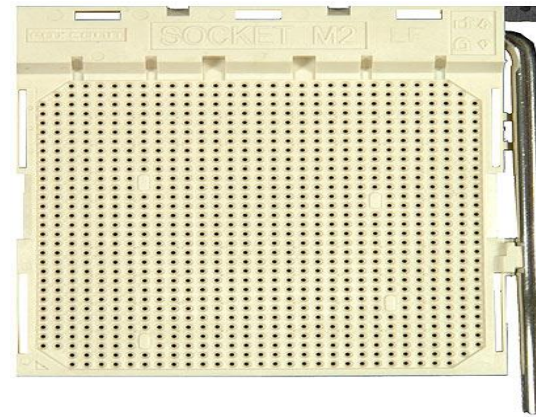
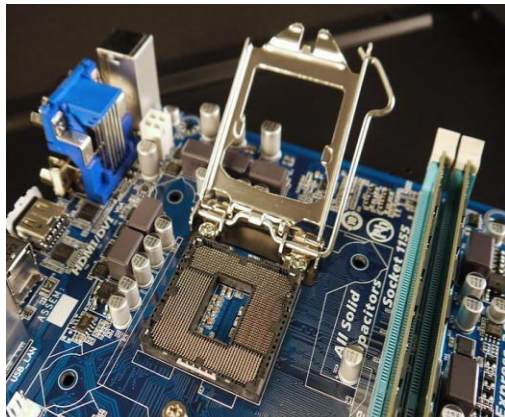
- **Pont southbridge:** - définit et commande le fonctionnement de tous les bus et dispositifs plus lents tels que: le bus PCI, l'interface PS/2 (ancien) pour le clavier et la souris, le port série, le port parallèle.
 - Certaines dispositifs sont souvent prises en charge par un contrôleur secondaire d'entrée/sortie et, dans ce cas, le southbridge fournit une interface à ce contrôleur.

❑ Chipset



❑ Socket

- Le **socket** est l'emplacement où se connecte le micro-processeur sur la carte mère
- Un support qui permet de relier le microprocesseur électriquement à la carte mère sans le souder.



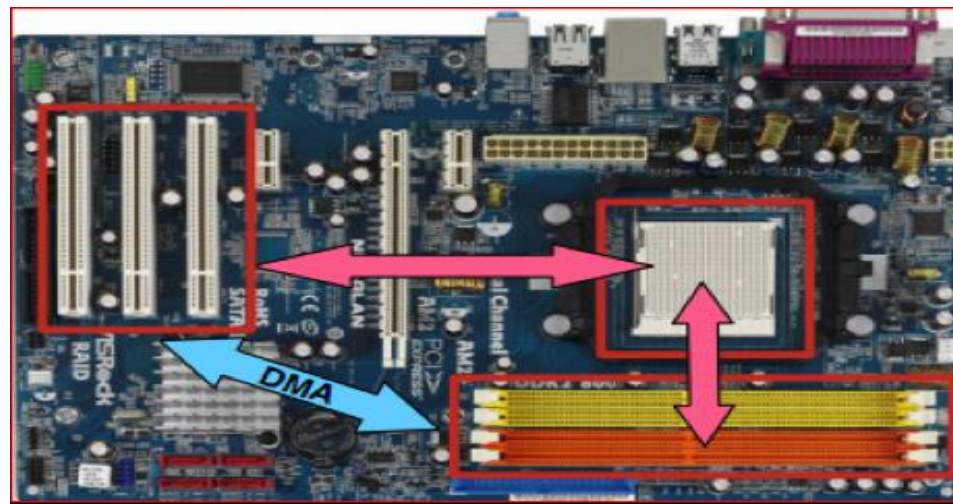
❑ Le support de mémoire

- Les cartes mères disposent de plusieurs supports mémoire permettant de connecter de la mémoire vive (la RAM) sous la forme de barrettes.



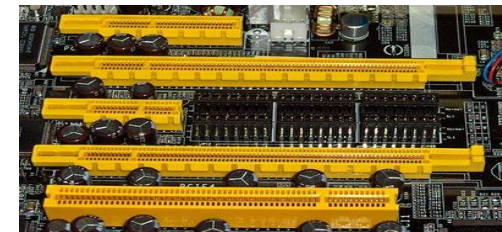
❑ Les ports internes

- Les port **PCI** (Peripheral Component Interconnect)
 - Il spécifie un bus local série (bus PCI express) et un connecteur qui sert à **connecter des cartes d'extension** (graphique, son, réseau) sur la carte mère d'un ordinateur.
 - Le bus PCI supporte le **DMA** (Direct Memory Access) qui permet à un périphérique de prendre le contrôle du bus pour **transférer des données directement** en mémoire.

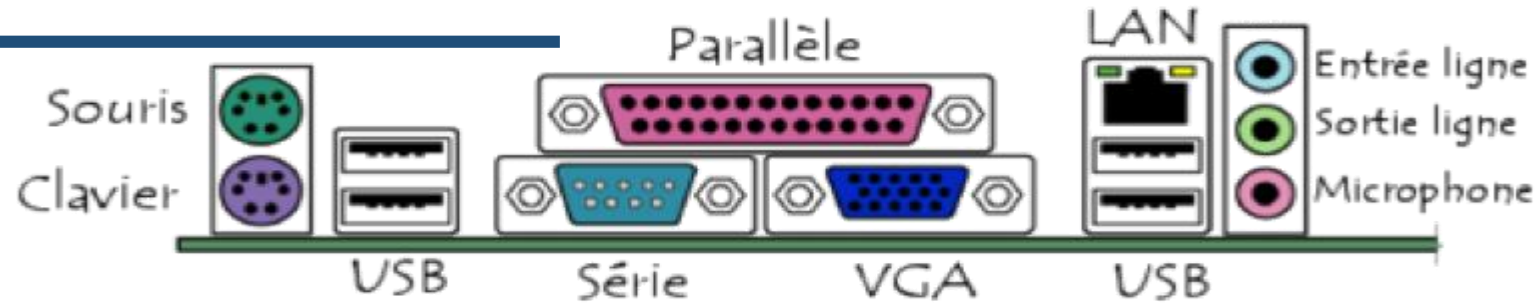


❑ Les ports internes

- **Le port ATA (Advanced Technology Attachment) (IDE)**
 - Ancienne norme pour connecter les périphériques de stockage interne (en voie de disparition)
- **Serial ATA (SATA)**
 - Propose des débits plus importants que la norme ATA
 - Il permet de connecter des disques durs ou des lecteurs optiques.
- **Le PCI Express**
 - Permet d'atteindre des débits de 250 Mo/s (version de base) ~ jusqu'à 8Go/s dans sa version x16 destinée à des périphériques nécessitant des bandes passantes très élevées (application graphique).



❑ Les ports externes



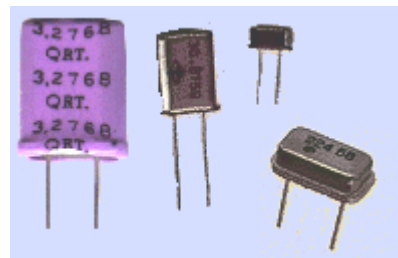
- **Le port USB**
 - Extrêmement connu pour son caractère universel (imprimante, scanner, disque dur, clavier, webcam, clé...).
 - USB 1.0, USB 2.0, USB 3.0
- **Le port eSATA**
 - Version externe de SATA
- **Les ports parallèles**
 - autorisent l'utilisation d'une imprimante
- **Les ports série**
 - Encore fréquemment utilisé pour connecter des appareils électroniques (Industrie)
 - Souvent désignés COM1, COM2...
- **Connexion réseau**
 - Filaire: RJ45



□ L'horloge

- Horloge temps réel (RTC = Real Time Clock) : circuit chargé de la synchronisation des signaux du système
- Elle est constituée d'un cristal de quartz qui, en vibrant, donne des impulsions (**tops d'horloge**) afin de cadencer le système.
- Fréquence de l'horloge (exprimée en **MHz**) : nombre de vibrations du cristal par seconde, c.à.d, **nombre de tops d'horloge émis par seconde**
- Plus la fréquence est élevée, plus le système peut traiter d'informations.

Exemples de quartz



❑ La pile du CMOS

- CMOS (Complementary Metal-Oxyde Semiconductor, ou BIOS CMOS) : circuit électronique qui **conserve certaines informations** sur le système t.q. l'heure, la **date système** et quelques **paramètres essentiels** du système.
- Le CMOS est **continuellement alimenté** par une pile ou une batterie située sur la carte-mère.

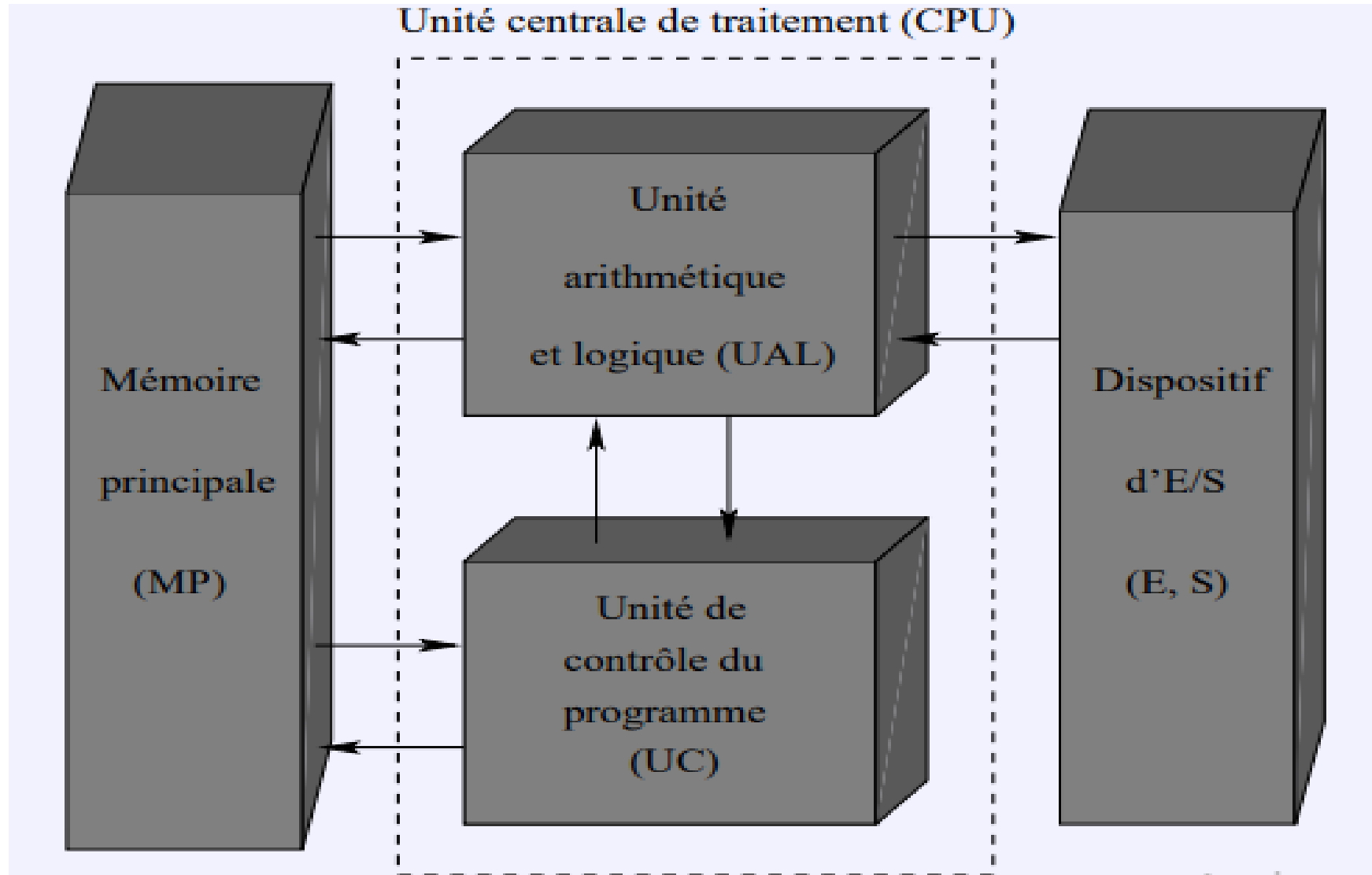




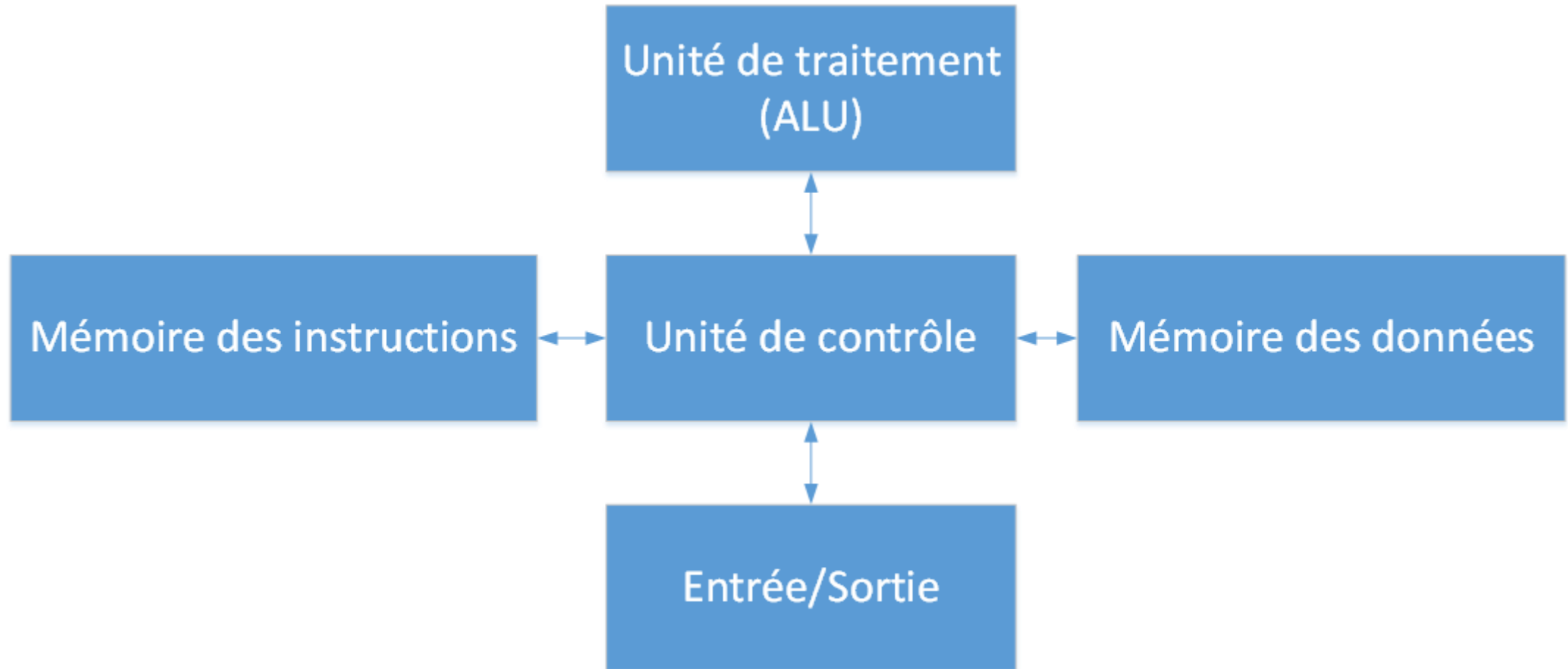
❑ Le BIOS

- **BIOS** (Basic Input/Output System) : programme basique servant d'interface entre le système d'exploitation et la carte-mère.
- **Le BIOS est stocké:**
 - dans une ROM (mémoire morte) et utilise les données contenues dans le CMOS pour connaître la configuration matérielle du système
 - dans une EEPROM (mémoire modifiable par impulsions électriques)
- Lorsque le système est mis sous-tension ou réamorcé (Reset), le BIOS fait l'inventaire du matériel présent dans l'ordinateur et effectue un test appelé **POST (Power-On Self Test)** afin de vérifier son bon fonctionnement
- La plupart des BIOS ont un « **setup** » (**programme de configuration**) qui permet de modifier la configuration basique du système.

□ Architecture de von Neumann



□ Architecture de Harvard

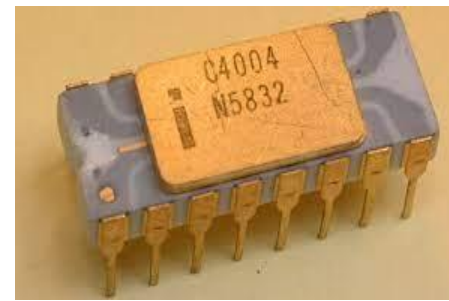


❑ Mémoire Principale

- Elle contient les instructions des programmes en cours d'exécution et les données associées à ce programme
- **Composée :**
 - **Mémoire vive (RAM)**
 - Lecture/écriture
 - chargée de stocker les données intermédiaires ou les résultats de calculs.
 - données sont perdues à la mise hors tension
 - **Mémoire morte (ROM)**
 - Lecture seule
 - Stocke les programmes et données de base de la machine ;
 - Lors d'une coupure du courant électrique, son contenu n'est pas perdu



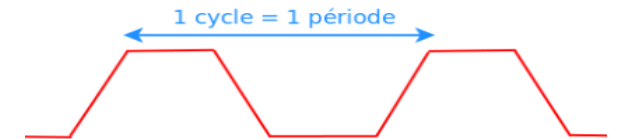
❑ Le processeur



- Jusqu'au début des années 1970, les différents composants électroniques, nécessaires au fonctionnement d'un processeur ne pouvaient pas tenir sur un seul circuit intégré.
- En 1971, la société Intel réussit à placer tous les composants qui constituent un processeur sur **un seul circuit intégré** => microprocesseur.
- Cette miniaturisation a permis :
 - **d'augmenter les vitesses de fonctionnement des processeurs**: la réduction des distances entre les composants ;
 - **de réduire les coûts**: remplacement de plusieurs circuits par un seul ;
 - **d'augmenter la fiabilité**: en supprimant les connexions entre les composants du processeur, on supprime l'un des principaux vecteurs de panne.
 - **de créer des ordinateurs bien plus petits** : les ordinateurs personnels
 - **de réduire la consommation énergétique**

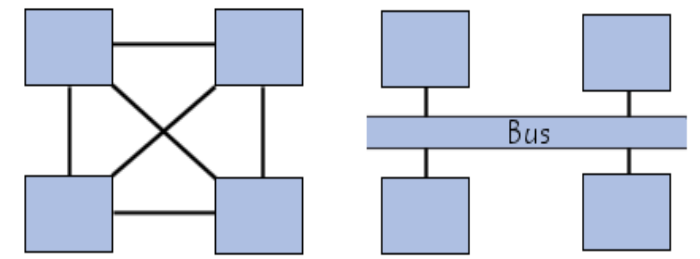
□ Le processeur

- Processeur (CPU = **Unité Centrale de Traitement**) : cerveau de l'ordinateur cadencé au rythme d'une horloge interne (ex: un ordinateur à 200 MHz possède une horloge envoyant 200 millions de battements par seconde)
- A chaque top d'horloge le processeur **exécute une action**, correspondant à une **instruction** ou une **partie d'instruction**
- L'**indicateur CPI** (Cycles Par Instruction) permet de représenter le nombre moyen de cycles d'horloge nécessaire à l'exécution d'une instruction
- **La puissance du processeur** = nombre d'instructions qu'il est capable de traiter par seconde exprimé en MIPS (Millions d'Instructions Par Seconde)
= fréquence du processeur / CPI



□ Le processeur: les unités fonctionnelles

- **Unité de contrôle:** lit les données arrivant, les décode puis les envoie à l'unité d'exécution
 - séquenceur (ou bloc logique de commande) : synchronise l'exécution des instructions au rythme d'une horloge (envoi des signaux de commande)
 - registre d'instruction : contient l'instruction en cours de traitement
 - compteur ordinal : contient l'adresse de la prochaine instruction à traiter
- **Unité d'exécution (ou unité de traitement) :** accomplit les tâches envoyés par UC
 - unité arithmétique et logique (UAL): effectue les opérations machines de l'ordinateur (opérations arithmétiques/logiques)
 - registre d'état : stocke des indicateurs sur l'état du système (retenue, dépassement, etc.)
 - registre accumulateur : stocke les résultats des opérations
- **Unité de gestion des bus (ou unité d'E/S) :** gère les flux d'informations entrant et sortant.



❑ Notion de bus

- Un ensemble de liaisons physiques (câbles, pistes de circuits imprimés, etc.) pouvant être exploitées en commun par plusieurs éléments matériels afin de communiquer.
- **Objectif** : réduire le nombre de voies nécessaires à la communication des différents composants.
- Un bus se décompose en 3 parties:
 - **le bus d'adresses** (bus d'adressage ou bus mémoire) transporte les adresses mémoire auxquelles le processeur souhaite accéder pour lire ou écrire une donnée.
 - **le bus de données** transporte les instructions en provenance ou à destination du processeur.
 - **le bus de commandes** (bus de commandes) transporte les ordres et les signaux de synchronisation en provenance de l'unité de commande et à destination de l'ensemble des composants matériels.

❑ Notion de bus

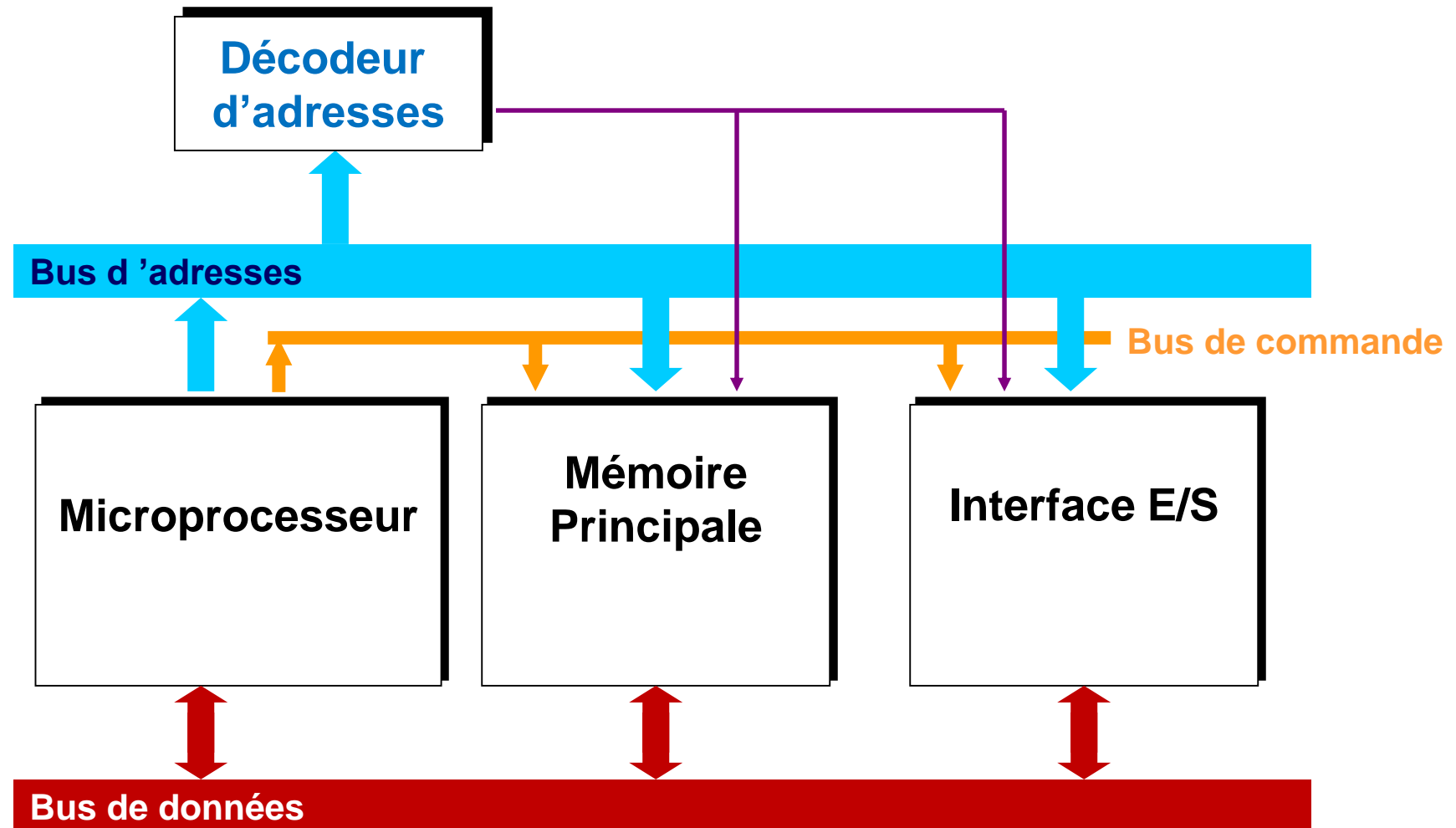
- On distingue généralement sur un ordinateur deux principaux bus :
 - **le bus système** (*bus interne, internal bus* ou *front-side bus*, noté *FSB*).
Le bus système permet au processeur de **communiquer** avec la mémoire centrale du système (RAM).
 - **le bus d'extension** (*bus d'entrée/sortie*) permet aux divers composants de la carte-mère (USB, série, parallèle, cartes branchées sur les connecteurs PCI, disques durs, lecteurs et graveurs de CD-ROM, etc.) de **communiquer** entre eux mais il permet surtout **l'ajout de nouveaux périphériques** grâce aux connecteurs d'extension (appelés **slots**) connectés sur le bus d'entrées-sorties.

□ Notion de bus

- Un bus est caractérisé par :
 - **Type** : parallèle ou série : Avec un bus parallèle, les données sont transmises plusieurs bits à la fois. Cependant, avec un bus série, les données sont transférées un bit à la fois.
 - **Largeur** : exprimée en nombre de bits ou d'octets : il s'agit du nombre de bits qui sont transférés en même temps
 - **Fréquence** : exprimée en hertz : elle indique la vitesse de transfert de l'information
- En fonction de la fréquence et de la largeur, on peut déterminer la bande passante ou débit qui s'exprime en Mo/s ou Go/s (ou Mbits/s, Gbits/s) :

$$\text{bande passante} = \text{fréquence} \times \text{largeur}$$

❑ Décodeur d'adresse



❑ Décodeur d'adresse

- La multiplication des périphériques autour du microprocesseur oblige la présence d'un décodeur d'adresse:
 - Il est chargé **d'aiguiller** les données présentes sur le bus de données provenant des multiples périphériques .
 - Le microprocesseur peut **communiquer** avec les différentes mémoires et les différents périphériques qui sont tous reliés sur le même **bus de données** et afin d'**éviter des conflits**, un seul composant doit être sélectionné à la fois.
 - Attribue à **chaque périphérique** une **zone d'adresse** et une **fonction** « décodage d'adresse » afin de fournir les signaux de sélection de chacun des composants

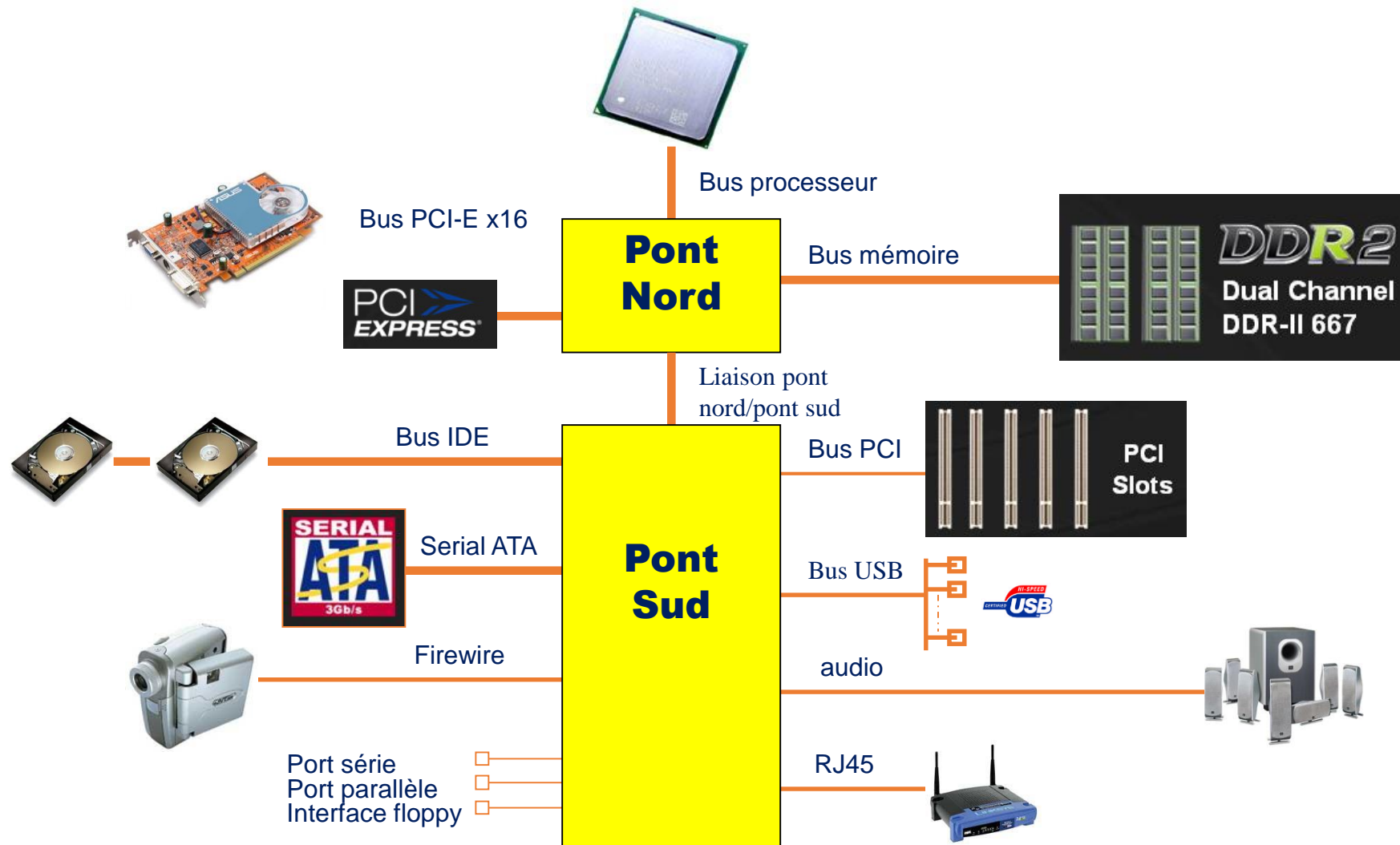
❑ Les périphériques d'entrée

- Permettent de réaliser la tâche d'entrée (**Input Operations**)
- Exemples :
 - **Clavier**
 - **Souris**
 - **Scanner**
 - **Microphone**
 - **Manette de jeu**

❑ Les périphériques de sortie

- Permettent de réaliser la tâche de sortie (**Output Operations**)
- Exemples :
 - Écran
 - Imprimante
 - Table traçante
 - Haut-parleur

□Récapitulatif



Les mémoires

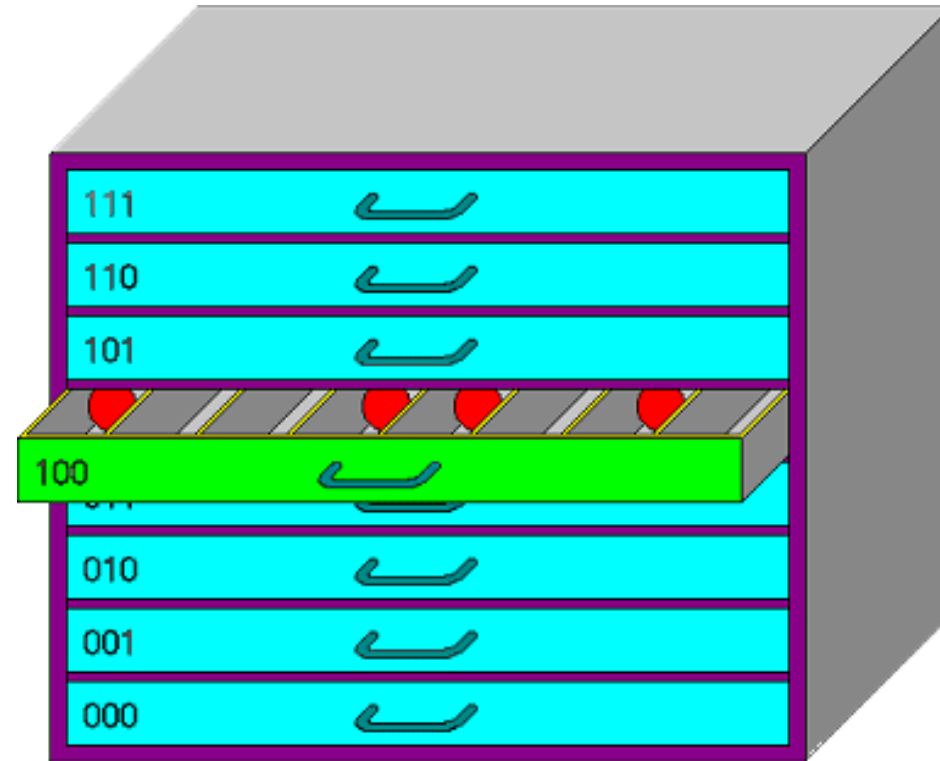
□ La mémoire: Définition

- Un dispositif permettant d'enregistrer, de conserver et de restituer des informations
- Permet le stockage des informations sous forme numérique
- Les informations stockées sont appelées des **données**
- Elles sont accessibles par l'intermédiaire de leur **adresses**
- **Applications:**
 - Carte à puce
 - Carte mémoire
 - Appareil multimédia
 - Ordinateur
 - Etc...

□ La mémoire: Définition

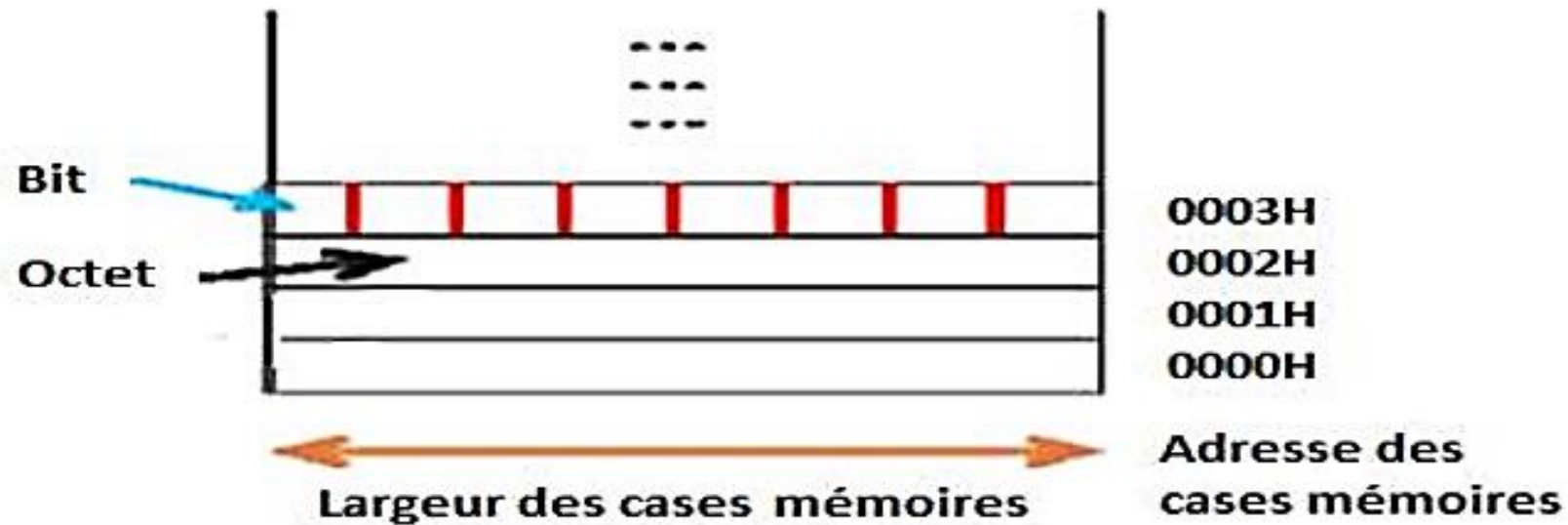
- La mémoire peut être vue comme **un ensemble de cellules** ou cases contenant chacune une information :
- Chaque case mémoire est repérée par un numéro d'ordre unique : **son adresse**. Elle est exprimée généralement en Hexadécimale.
- Chaque case mémoire est divisée en emplacements de **taille fixe** (par exemple 8 bits)
- Les opérations permises sur la **mémoire** sont les opérations de **lecture** et **d'écriture**.

□ La mémoire

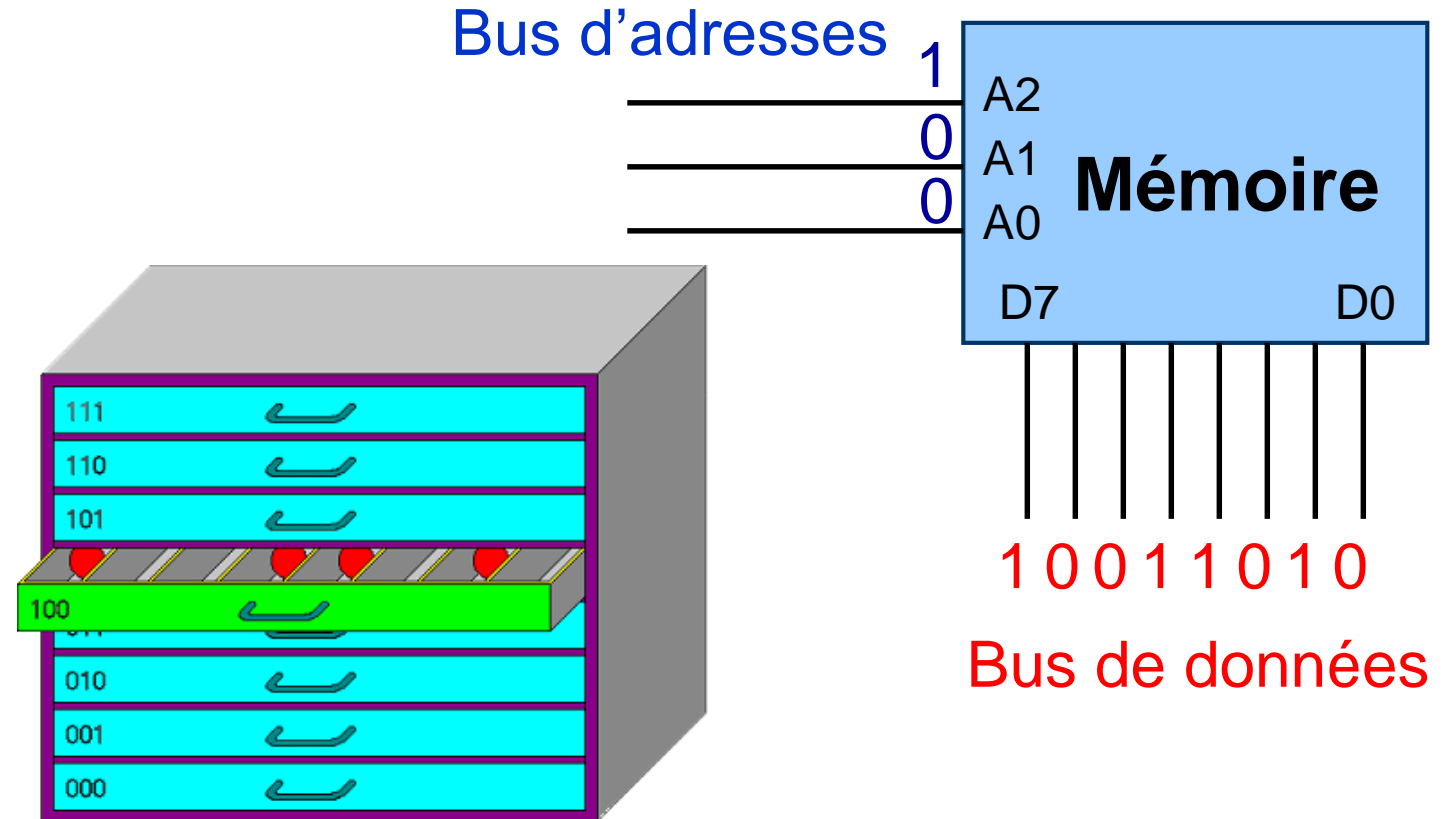


□ La mémoire

- Le nombre de fils d'adresses d'une mémoire définit le nombre de cases mémoires qu'elle comprend.
- Le nombre de fils de données définit la taille des données que l'on peut sauvegarder dans chaque case mémoire.



❑ La mémoire

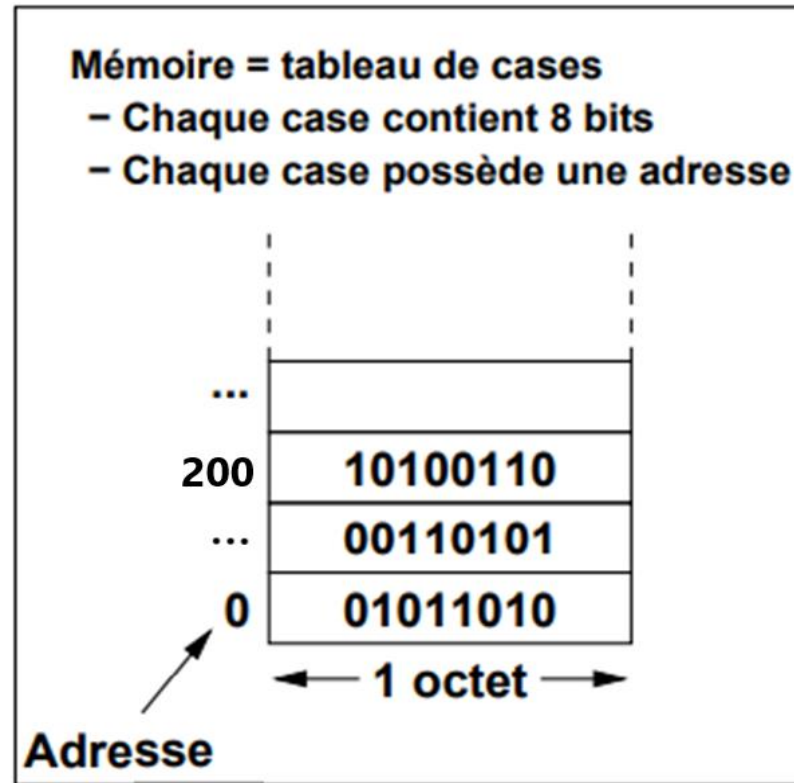


Question: Quel est le nombre minimal des fils nécessaire pour adresser cette mémoire ?

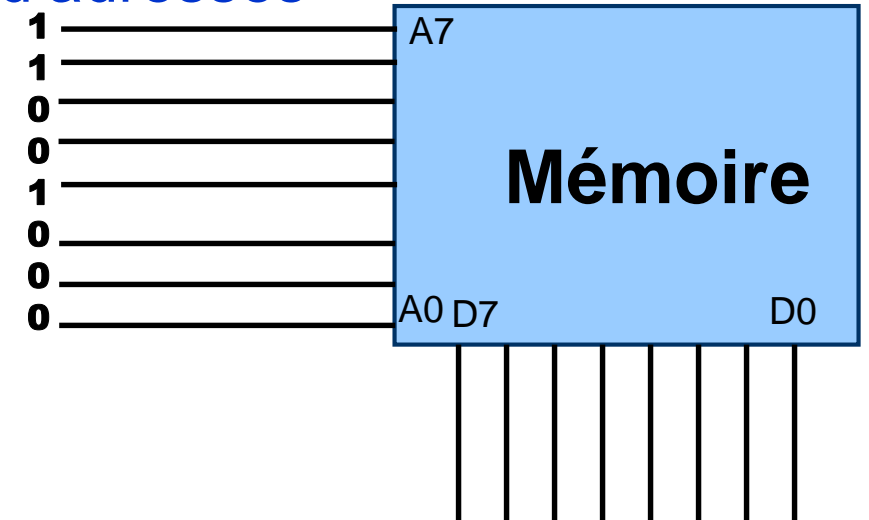
Réponse: 0 à 7 emplacements == 8 emplacements. Alors $8 = 2^3$

Donc: 3 fils sont nécessaires

❑ La mémoire



Bus d'adresses



1 0 1 0 0 1 1 0
Bus de données

Question: Supposant que le processeur veut lire le 3^{ème} emplacement, Quel sera le contenu du bus d'adresse ainsi que le bus de données?

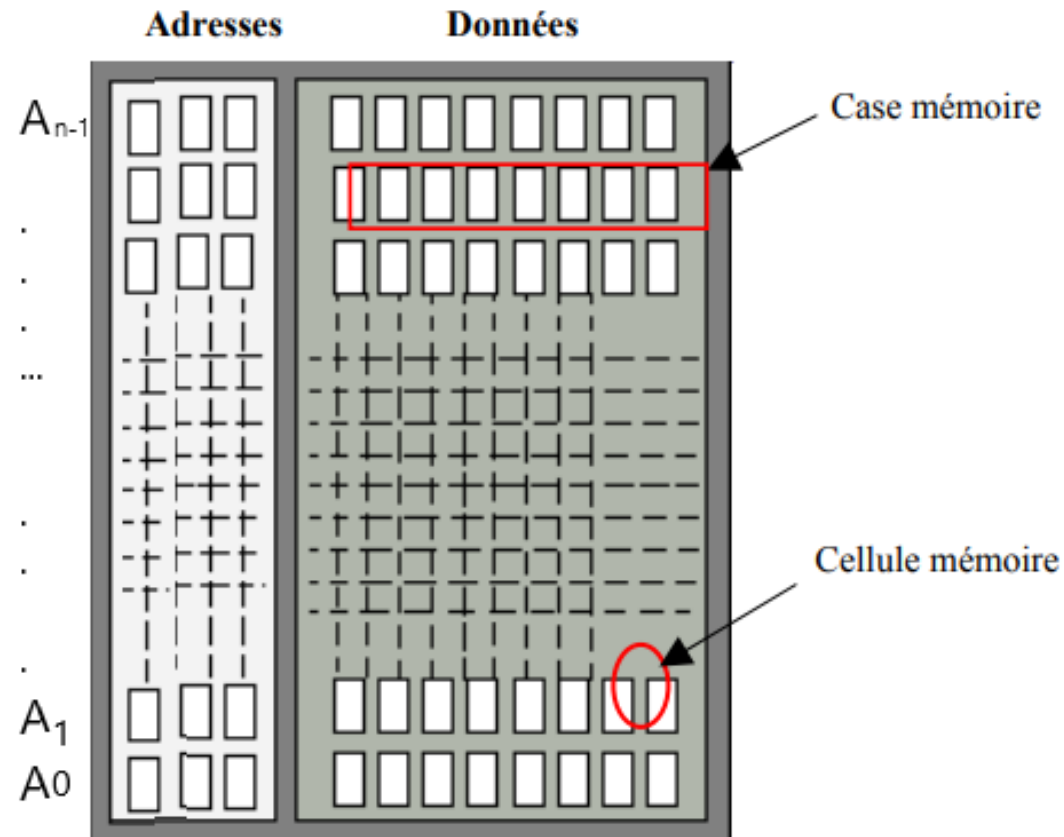
Réponse: Bus Adresse: 11001000
Bus données: 10100110

□ La mémoire

- Si les adresses des cases mémoires sont codées sur **n bits**, il est possible de référencer au plus 2^n cases mémoire.
 - **Exemple** : si les adresses des cases mémoires sont codées sur 4 bits, il est alors possible d'avoir $2^4 = 16$ cases mémoire
- **Chaque case est remplie par un mot de données:**
 - Un mot est l'unité d'information accessible en **une seule opération de lecture** (sa taille varie en fonction de la machine).
 - Un mot représente un **regroupement de bits**
 - Sa longueur est toujours une **puissance de 2**

□ La mémoire

- Si les adresses des cases mémoires sont codées sur **n bits**, il est possible de référencer au plus 2^n cases mémoire.



❑ Classification des mémoires

Les mémoires peuvent être classées en trois catégories selon la technologie utilisée :

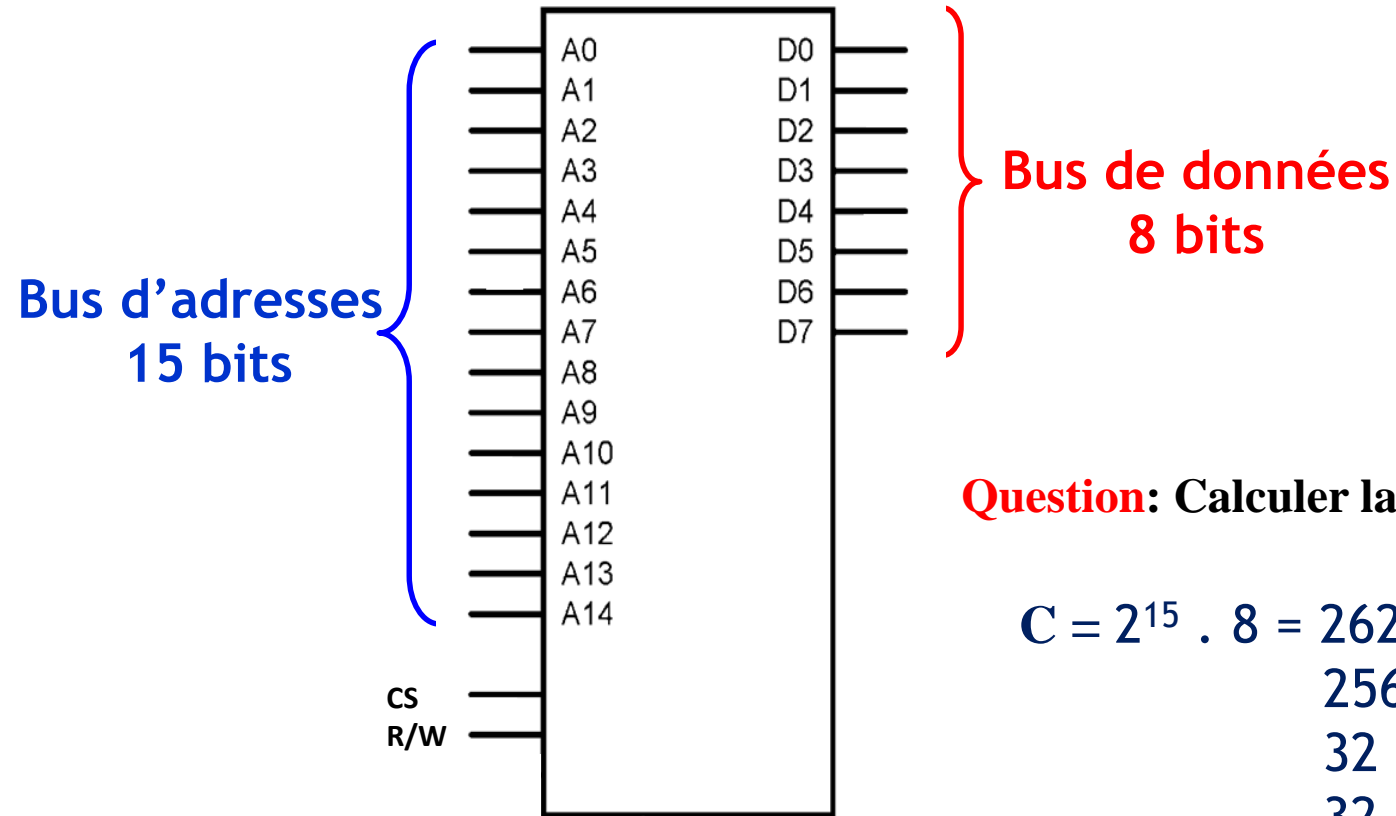
- **Mémoire à semi-conducteur** (mémoire centrale, ROM,.....) : très rapide mais de taille réduite.
- **Mémoire magnétique** (disque dur,...) : moins rapide mais stocke un volume d'informations très grand.
- **Mémoire optique** (DVD, CDROM,..)

❑ Caractéristiques de la mémoire

- **Format des données** : c'est le nombre de bits que l'on peut mémoriser par case mémoire. On dit aussi que c'est la largeur du mot mémorisable.
- **Capacité**: volume global d'informations (en bits) que la mémoire peut stocker.
 - Si nous avons m bits dans chaque case mémoire et s'il existe 2^n cases mémoire alors la capacité est : $C = m * 2^n$
 - **Octet** : 1 Octet = 8 bits
 - **kilo-octet (Ko)** : 1 kilo-octet (KO) = 1024 octets = 2^{10} octets
 - **Méga-octet (Mo)** : 1 Méga-octet (MO) = 1024 KO = 2^{20} octets
 - **Géga-octet (Go)** : 1Géga-octet (GO) = 1024 MO = 2^{30} octets
 - **Téra-octet (To)** : 1 téra-octet (To) = 1024 Go = 2^{40} octets

❑ Caractéristiques de la mémoire

▪ Exemple : capacité mémoire



Question: Calculer la capacité C?

$$C = 2^{15} \cdot 8 = 262\,144 \text{ bits}$$

256 k bits

32 768 octets

32 ko

32 768 mots de 8 bits

Annotations: A blue arrow points from 262 144 bits to 256 k bits. A red arrow points from 32 768 octets to 32 ko. An orange arrow labeled /1024 points from 256 k bits to 32 768 octets. A red arrow labeled /8 points from 32 768 octets to 32 ko.

❑ Caractéristiques de la mémoire

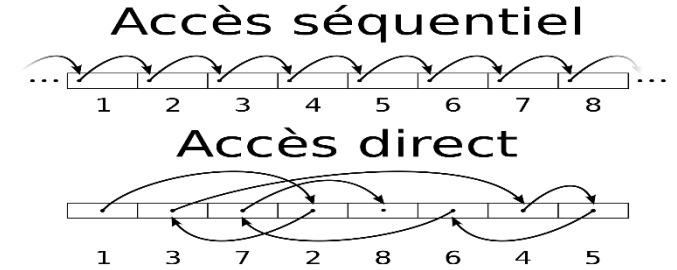
- **Temps d'accès:** c'est le temps qui s'écoule entre l'instant où a été lancée une opération de lecture/écriture en mémoire et l'instant où la première information est disponible sur le bus de données.
- **Temps de cycle :** L'intervalle minimum qui doit séparer deux demandes successives de lecture ou d'écriture.

❑ Caractéristiques de la mémoire

- **Débit:** C'est le nombre maximum d'informations lues ou écrites par seconde.

On parle aussi de la **bande passante**

- **Volatilité** : elle caractérise la permanence des informations dans la mémoire.
L'information stockée est volatile si elle risque d'être altérée (effacée) par un défaut d'alimentation électrique et non volatile (ou **permanente**) dans le cas contraire



□ Mode d'accès à la mémoire

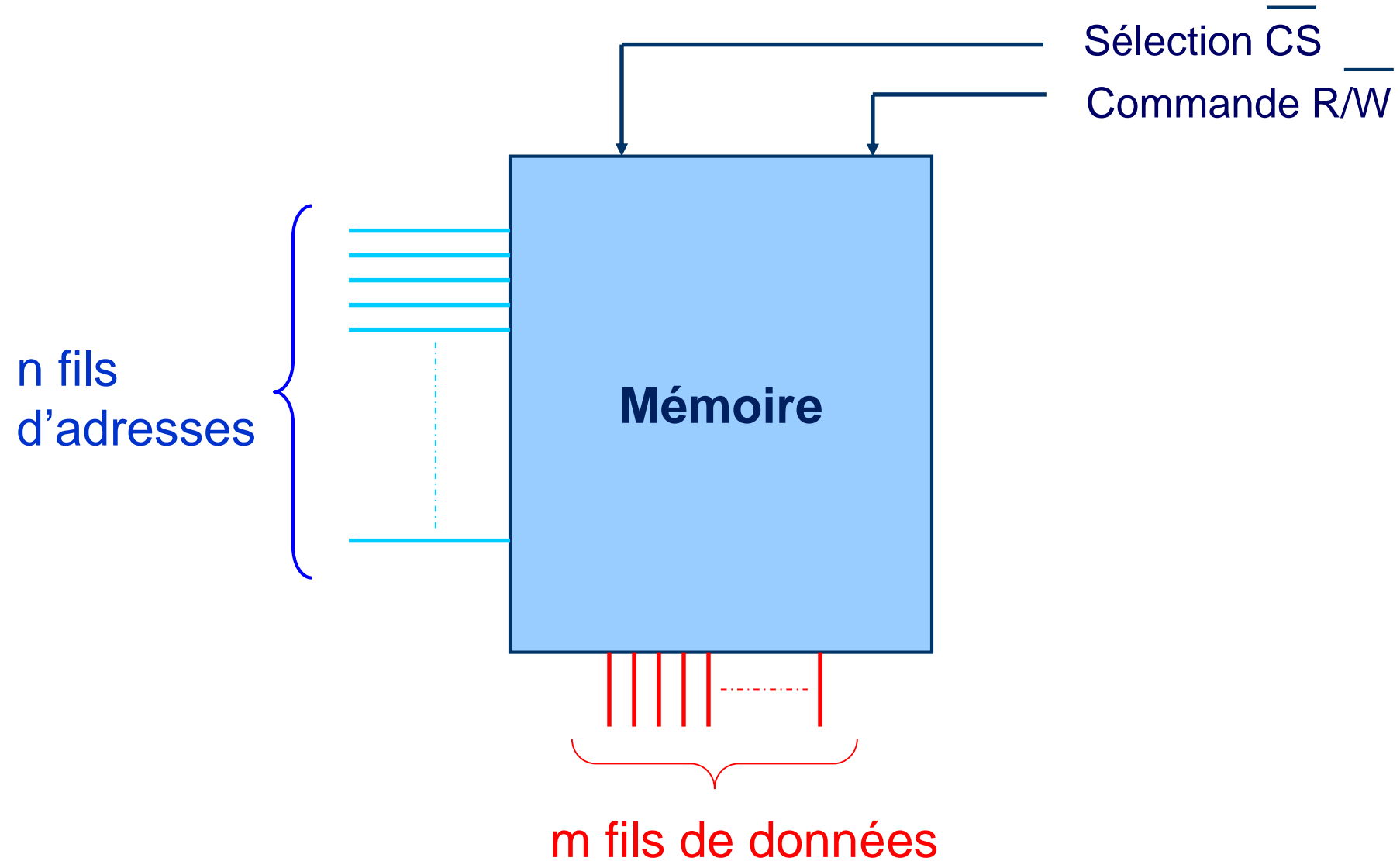
- **Direct:** Chaque information a une adresse propre. On peut accéder directement à chaque adresse
 - **Exemple :** mémoire centrale d'un ordinateur
- **Séquentiel:** Pour accéder à une information on doit parcourir toutes les informations précédentes (Accès lent)
 - **Exemple :** bandes magnétiques
- **Semi-séquentiel:** Intermédiaire entre séquentiel et direct
 - **Exemple :** disque dur: Accès direct au cylindre et Accès séquentiel au secteur sur un cylindre
- **Associatif/par le contenu :** Une information est identifiée par une clé. On accède à une information via sa clé
 - **Exemple :** mémoire cache



□ La mémoire: les entrées de commandes

- Un boitier mémoire comprend en plus des fils d'adresses et de données **les entrées de commandes**:
 - Une entrée de **commande (R/W)** qui permet de définir le type d'action que l'on effectue avec la mémoire (lecture/écriture).
 - Une entrée de **sélection CS (sélection du circuit)** qui permet la sélection de la mémoire correspondante

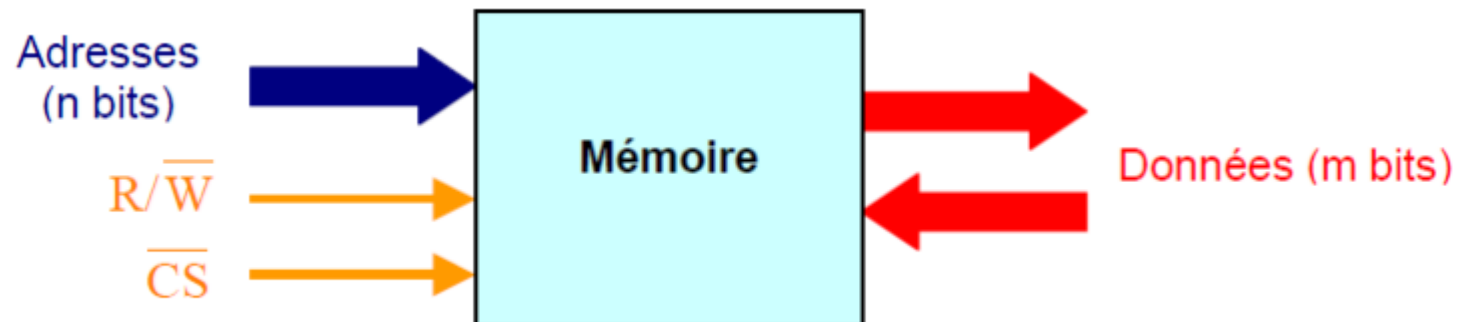
❑ La mémoire: les entrées de commandes



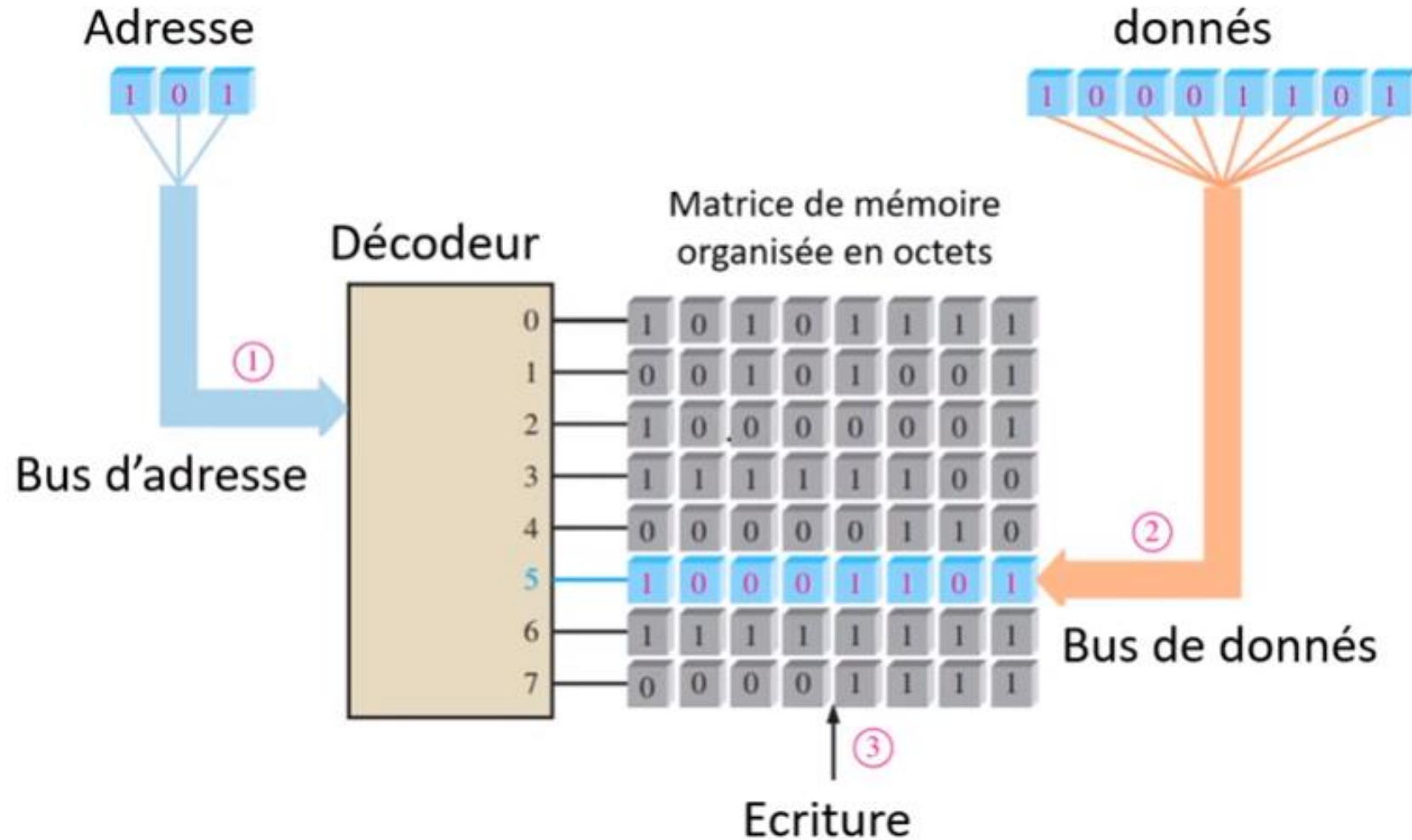
❑ Fonctionnement de la mémoire

Une opération de lecture ou d'écriture de la mémoire suit toujours le même cycle :

1. **Choisir l'adresse en mémoire** qui donne l'accès à un emplacement mémoire
2. **Choisir une opération** de lecture ou d'écriture (R/\overline{W})
3. **Sélectionner la mémoire** avec le bit Chip Select ($\overline{CS} = 0$)
4. **Effectuer l'opération** (lecture ou écriture)

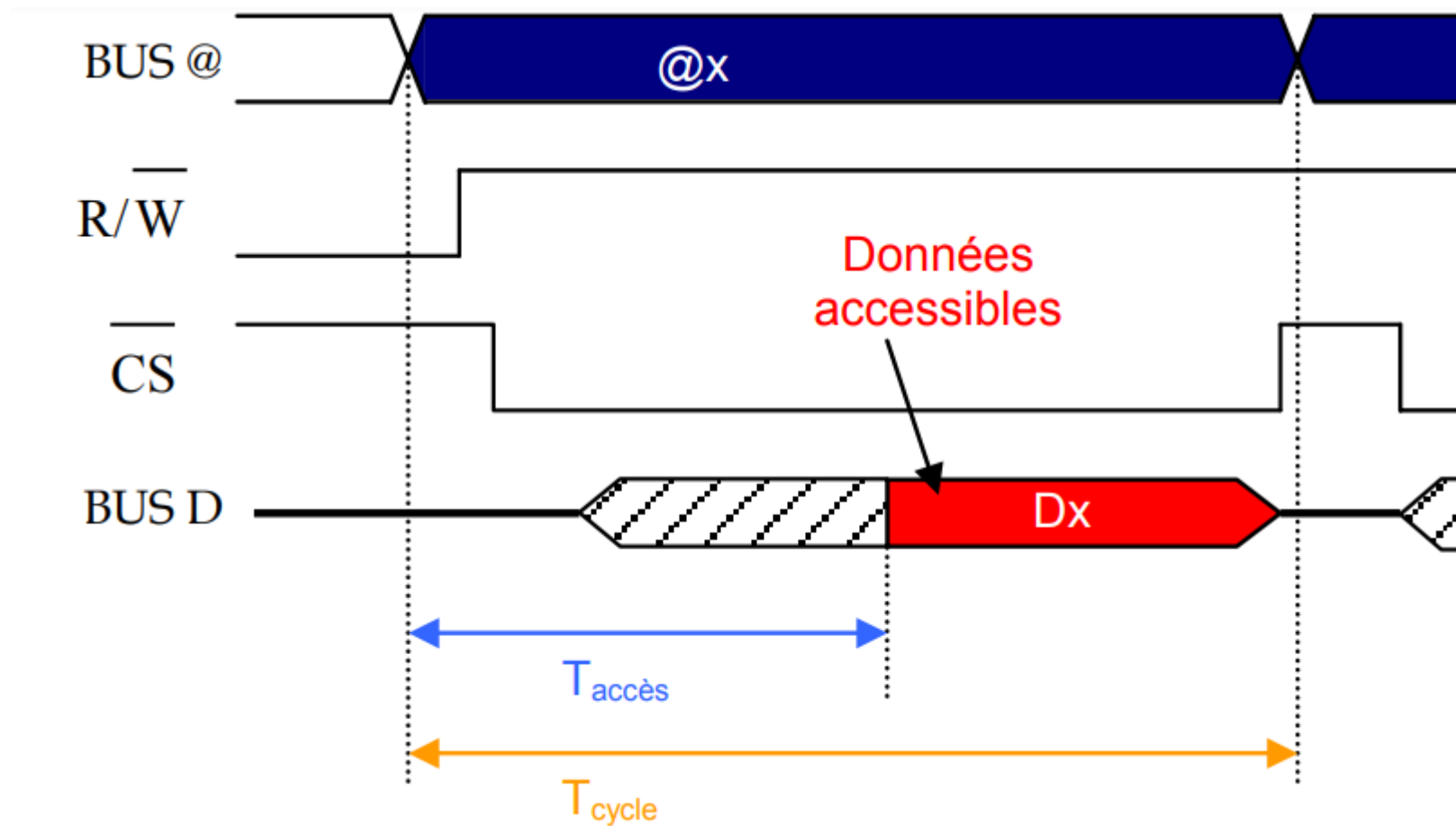


❑ Fonctionnement de la mémoire



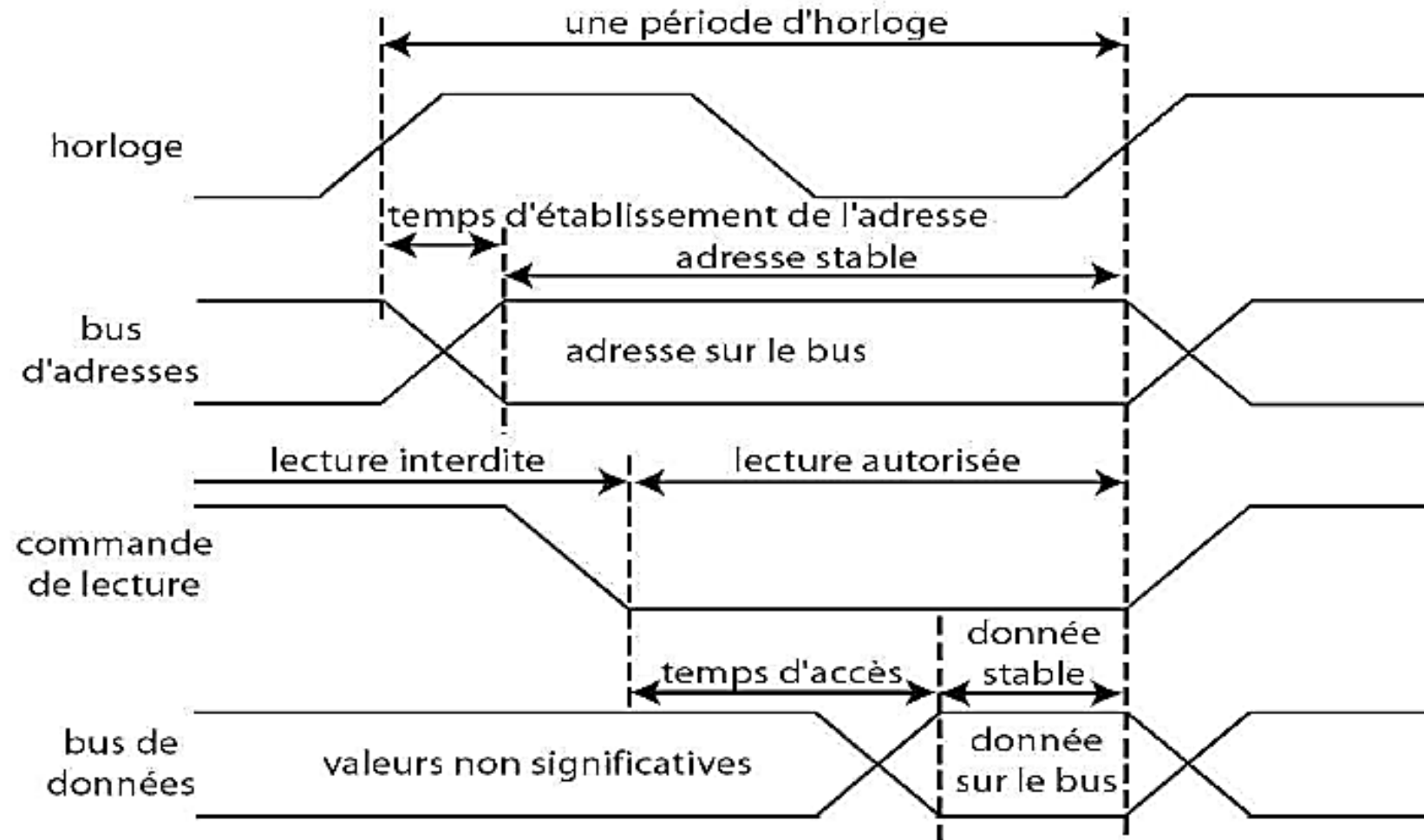
□ Fonctionnement de la mémoire

- Exemple : cycle de lecture



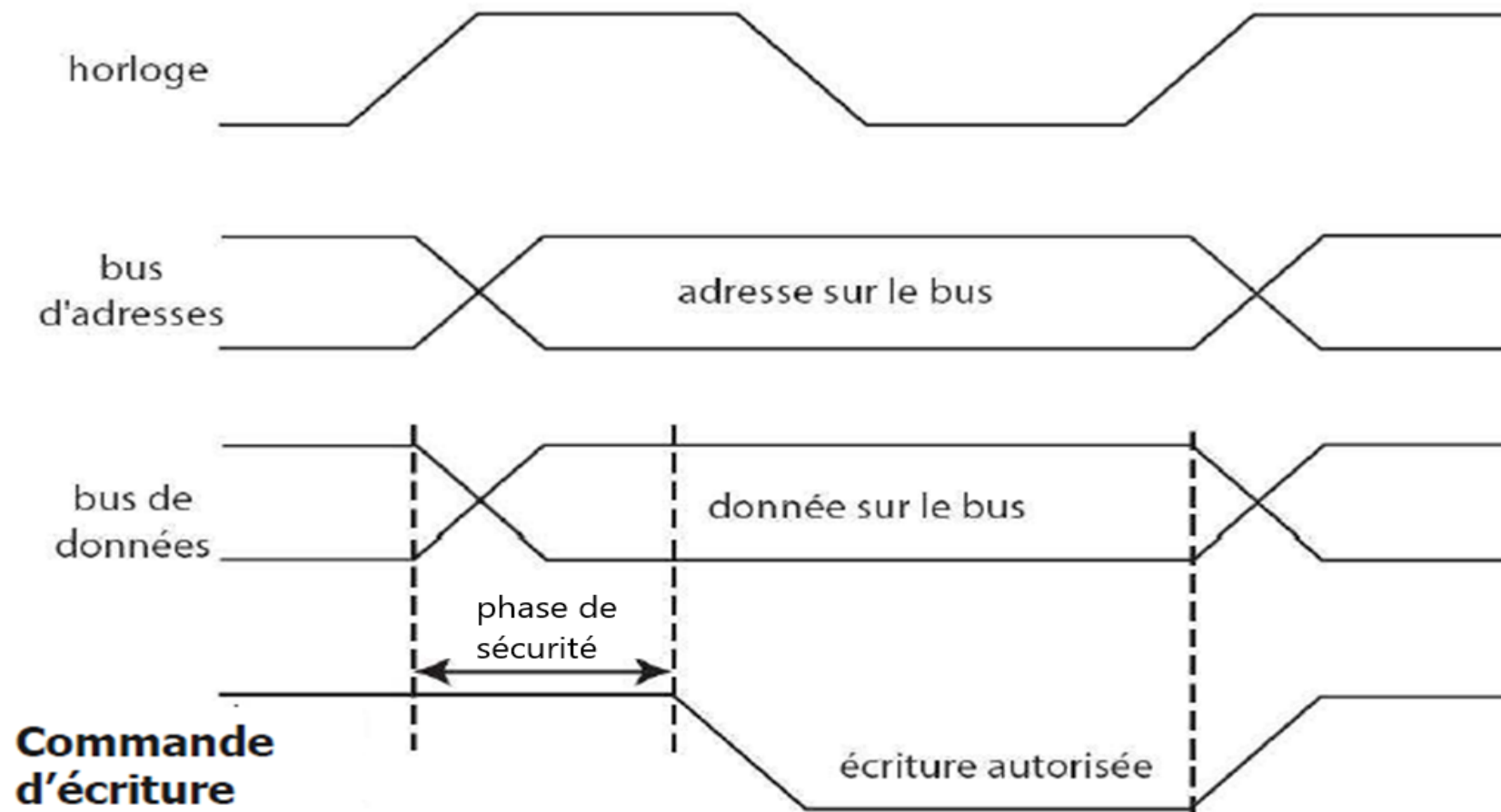
❑ Fonctionnement de la mémoire

▪ Chronogramme de lecture



□ Fonctionnement de la mémoire

▪ Chronogramme d'écriture

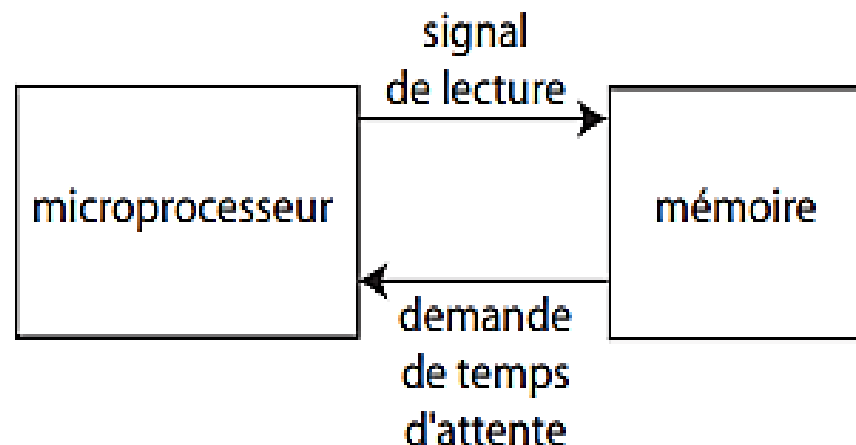


❑ Fonctionnement de la mémoire

- Temps d'attente

Si le temps d'accès d'une mémoire est supérieur à une période d'horloge (**mémoire lente**), le microprocesseur peut accorder à la mémoire un **temps supplémentaire** (une ou plusieurs périodes d'horloge), à la demande de celle-ci.

Ce temps supplémentaire est appelé **temps d'attente**

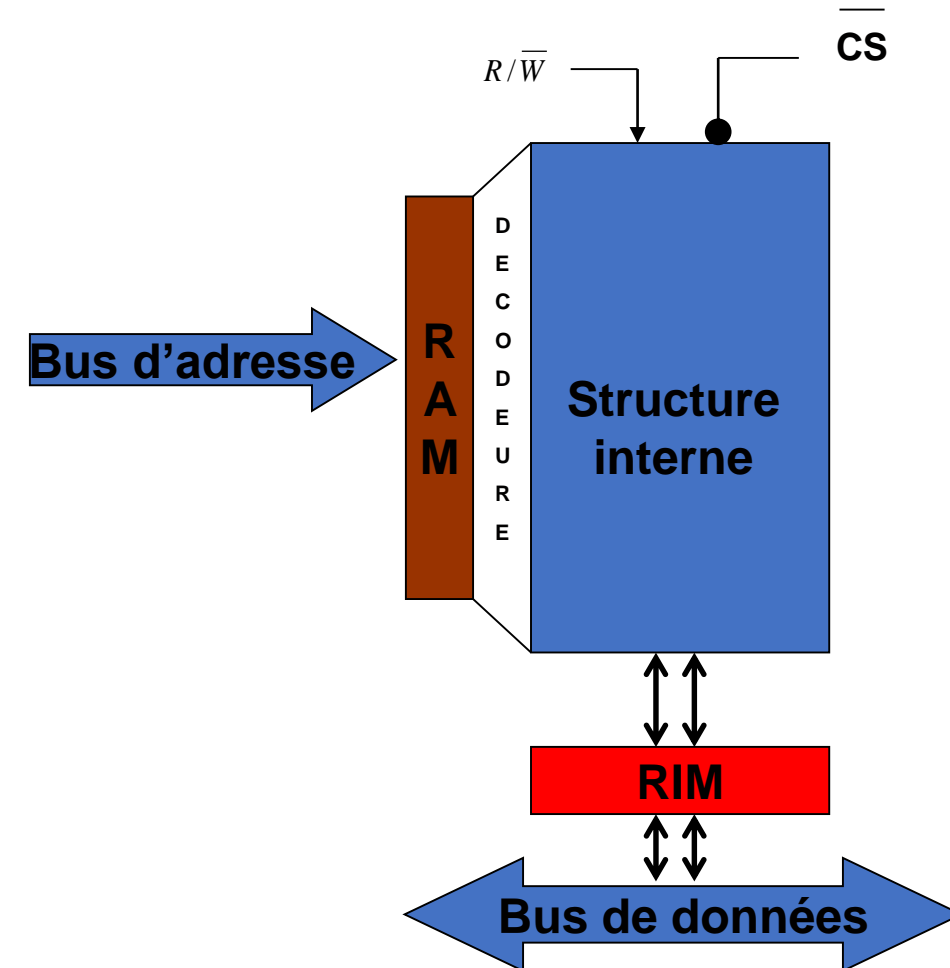


□ Mémoire centrale (MC)

- Structure physique
- Conception
- Exemples
- Exercice

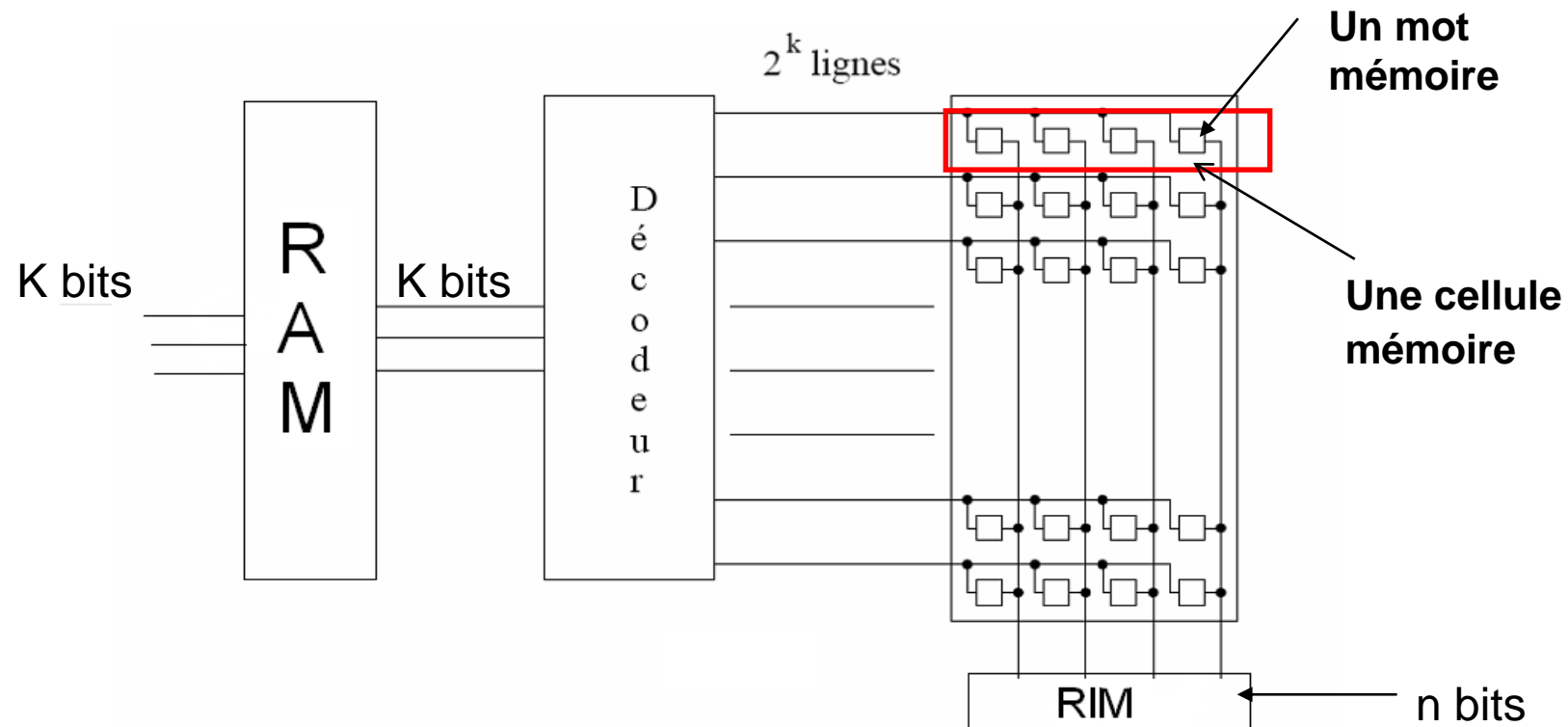
❑ Structure physique d'une mémoire centrale

- ❖ **RAM** (Registre d'adresse Mémoire) : ce registre stock l'adresse du mot à lire ou à écrire.
- ❖ **RIM** (Registre d'information mémoire) : stock l'information lu à partir de la mémoire ou l'information à écrire dans la mémoire.
- ❖ **Décodeur** : permet de sélectionner un mot mémoire.
- ❖ **R/W** : commande de lecture/écriture, cette commande permet de lire ou d'écrire dans la mémoire (si $R/W=1$ alors lecture sinon écriture)
- ❖ $\overline{CS}=0$ le boîtier est sélectionné
- ❖ $\overline{CS}=1$ le boîtier n'est pas sélectionné
- ❖ Bus d'adresses de taille **k bits**
- ❖ Bus de données de taille **n bits**



❑ Comment sélectionner un mot mémoire ?

- ❖ Lorsque une adresse est **chargée** dans le registre **RAM**, le décodeur va recevoir la même information que celle du RAM.
- ❖ A la sortie du décodeur nous allons avoir **une seule sortie** qui est active → Cette sortie va nous permettre de sélectionner **un seul mot mémoire**.



❑ Comment calculer la capacité d'une MC ?

- Soit k la taille du bus d'adresses (taille du registre RAM)
- Soit n la taille du bus de données (taille du registre RIM ou la taille d'un mot mémoire)
- On peut exprimer la capacité de la mémoire centrale soit en nombre de **mots mémoire** ou en **bits** (octets, kilo-octets,...)
 - La capacité ou Taille Mémoire (TM) = Nombre de Mots * Taille du Mot
 - La capacité = 2^k **Mots mémoire**
 - La capacité = $2^k * n$ bits

▪ Exemple :

Dans une mémoire la taille du registre RAM $k=14$ et la taille du registre RIM $n=4$.
Calculer la capacité de cette mémoire ?

- $C = 2^{14} = 16384$ Mots de 4 bits
- $C = 2^{14} * 4 = 65536$ Bits = 8192 Octets = 8 Ko

□ Comment lire une information ?

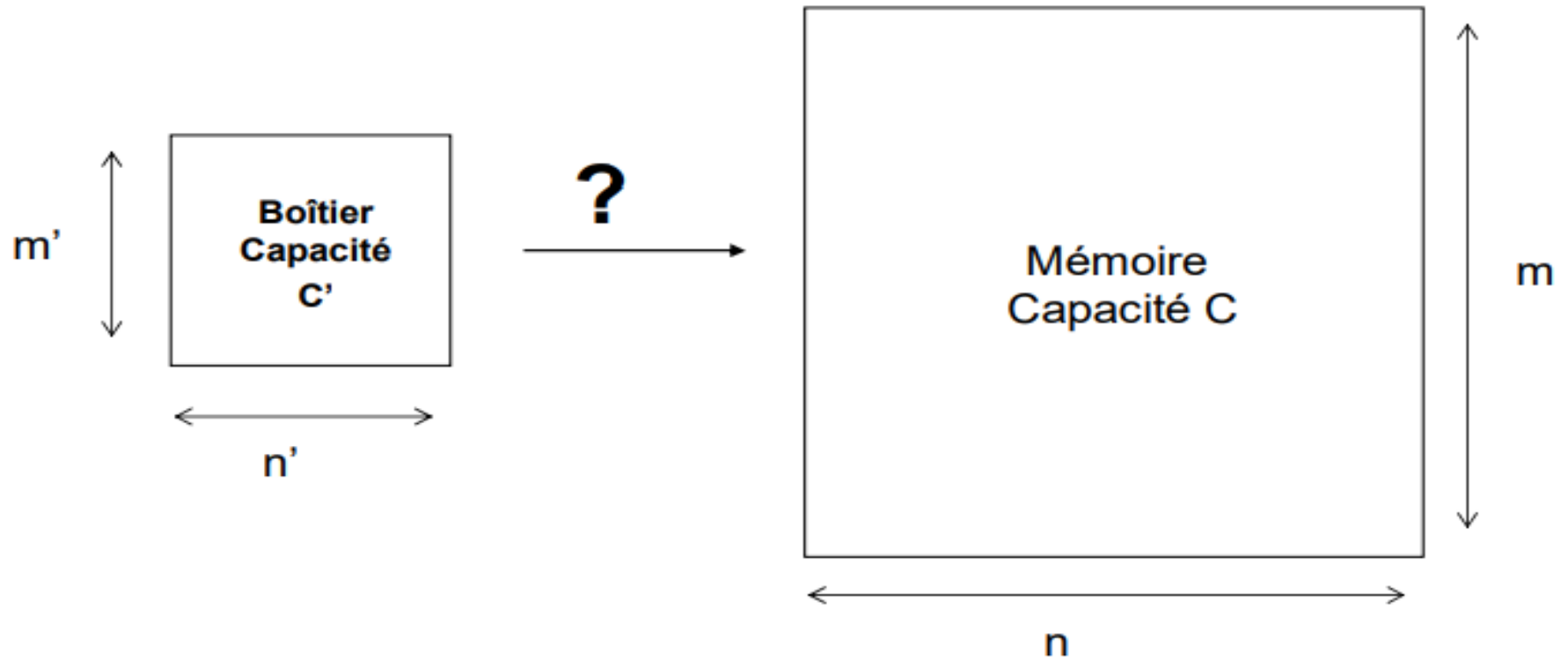
- Pour lire une information en mémoire centrale il faut effectuer les opérations suivantes:
 - **Charger** dans le registre RAM l'**adresse** du mot à lire.
 - **Lancer la commande de lecture** ($R/W=1$)
 - L'information est disponible dans le registre RIM au bout d'un certain temps
(**temps d'accès**)

□ Comment écrire une information ?

- Pour écrire une information en MC il faut effectuer les opérations suivantes:
 - **Charger** dans la RAM l'**adresse** du mot où se fera l'écriture.
 - **Placer** dans le RIM l'information à écrire.
 - **Lancer la commande d'écriture** pour transférer le contenu du RIM dans la mémoire.

❑ Problème ?

Comment réaliser une mémoire de **capacité C** à partir de boîtiers (des circuits) de **petite taille**?



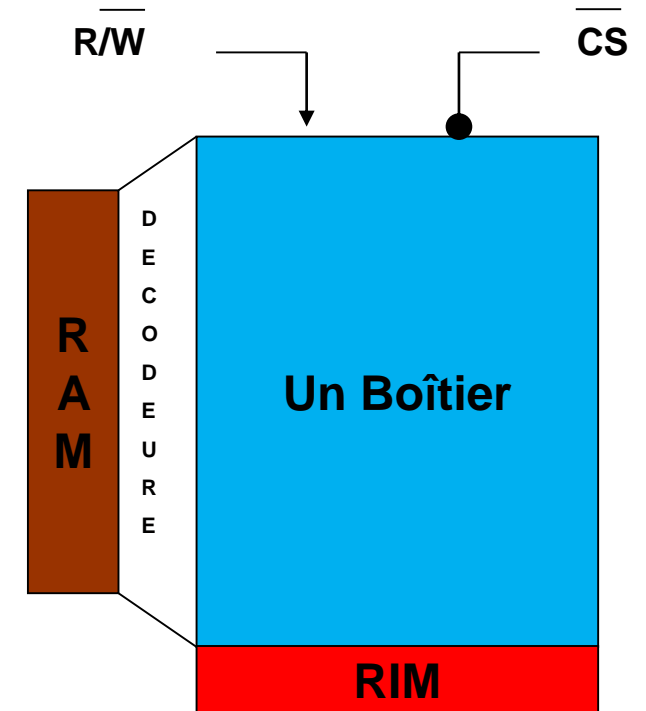
□ Structure d'un boîtier

Un boîtier possède la même structure qu'une mémoire (RAM, RIM, ...) en plus de la commande CS.

CS (Chip Select) : c'est une commande en logique négative qui permet de sélectionner (activer) un boîtier.

CS=0 le boîtier est sélectionné

CS=1 le boîtier n'est pas sélectionné



□ Solution

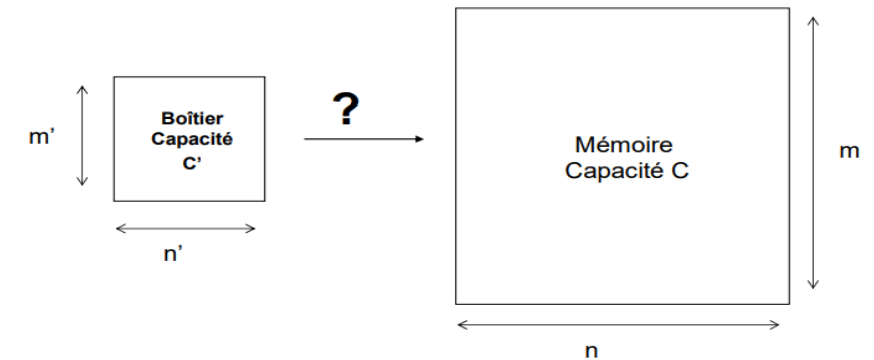
- Soit M une mémoire de capacité C, tel que **m** est le nombre de mot et **n** la taille d'un mot.
- Soit M' un boîtier de capacité C' , tel que m' le nombre de mot et n' la taille d'un mot.
- On suppose que $C > C'$ ($m \geq m'$, $n \geq n'$)
- Quel est le **nombre de boîtiers** M' nécessaire pour réaliser la mémoire M ?

Pour connaître le nombre de boîtiers nécessaire , il faut calculer les deux facteurs suivants :

$$\diamond P = m/m'$$

$$\diamond Q = n/n'$$

$$\checkmark \text{ Nombre de boîtiers nécessaire} = \mathbf{P \times Q}$$



❑ Solution suite

- **P** : permet de déterminer le nombre de boîtiers M' nécessaire pour obtenir le **nombre de mots** de la mémoire M (extension lignes).
- **Q** : permet de déterminer le nombre de boîtier M' nécessaire pour obtenir la **taille de mot** de la mémoire M (extension mots ou extension colonnes).
- **PxQ** donne le nombre totale de boîtiers nécessaire pour réaliser la mémoire M .
- Pour sélectionner les boîtiers on utilise **les bits de poids forts** d'adresses.
Si P est le facteur d'extension lignes alors on prend k bits tel que $P=2^k$.
- Les autres bits d'adresses restants sont utilisés pour sélectionner un mot dans un boîtier.

□ Exemple 1:

- Réaliser une mémoire de 1Ko (la taille d'un mot est de 8 bits) en utilisant des boîtiers de taille 256 mots de 8 bits ?

- **Solution :**

- $(m,n)=(1024,8) \rightarrow$ taille du bus d'adresses est de 10 bits $A^9_0(A9...A0)$,
taille du bus de données est de 8 bits $D^7_0(D7....D0)$
- $(m',n')=(256,8) \rightarrow$ taille du bus d'adresses est de 8 bits $(A7'...A0')$,
taille du bus de données est de 8 bits $(D7'....D0')$

Calculer les deux facteurs d'extension lignes et colonnes :

- $P = m/m' = 1024/256 = 4$ (**extension lignes**)
- $Q = n/n' = 8/8 = 1$ (**extension colonnes**)

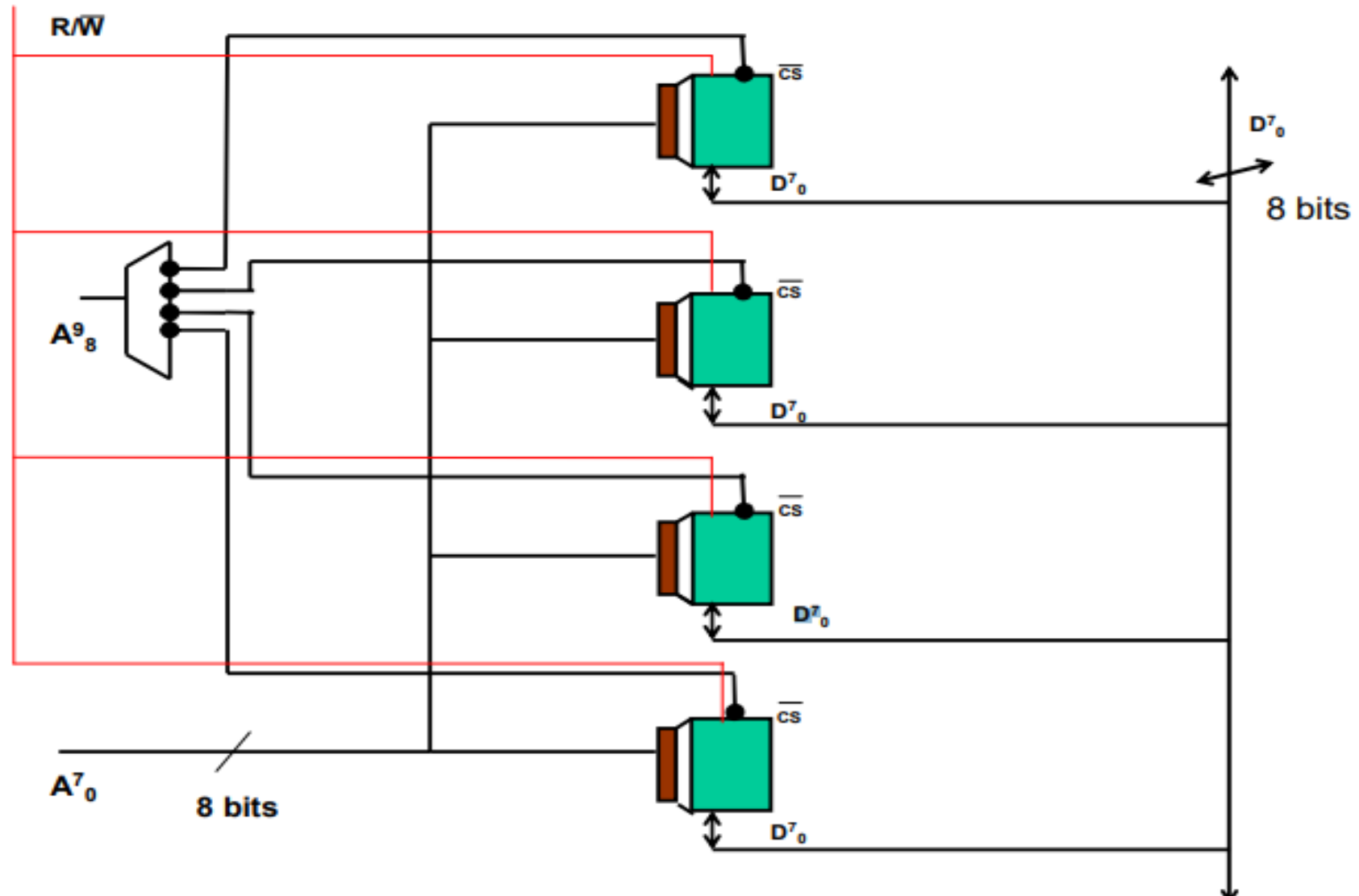
Le nombre totale de boîtiers $P.Q=4$

On remarque que $P=4=2^2$

donc les **deux** bits A9 et A8 seront réservé pour adresser le boîtier

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
----	----	----	----	----	----	----	----	----	----

❑ Exemple 1:



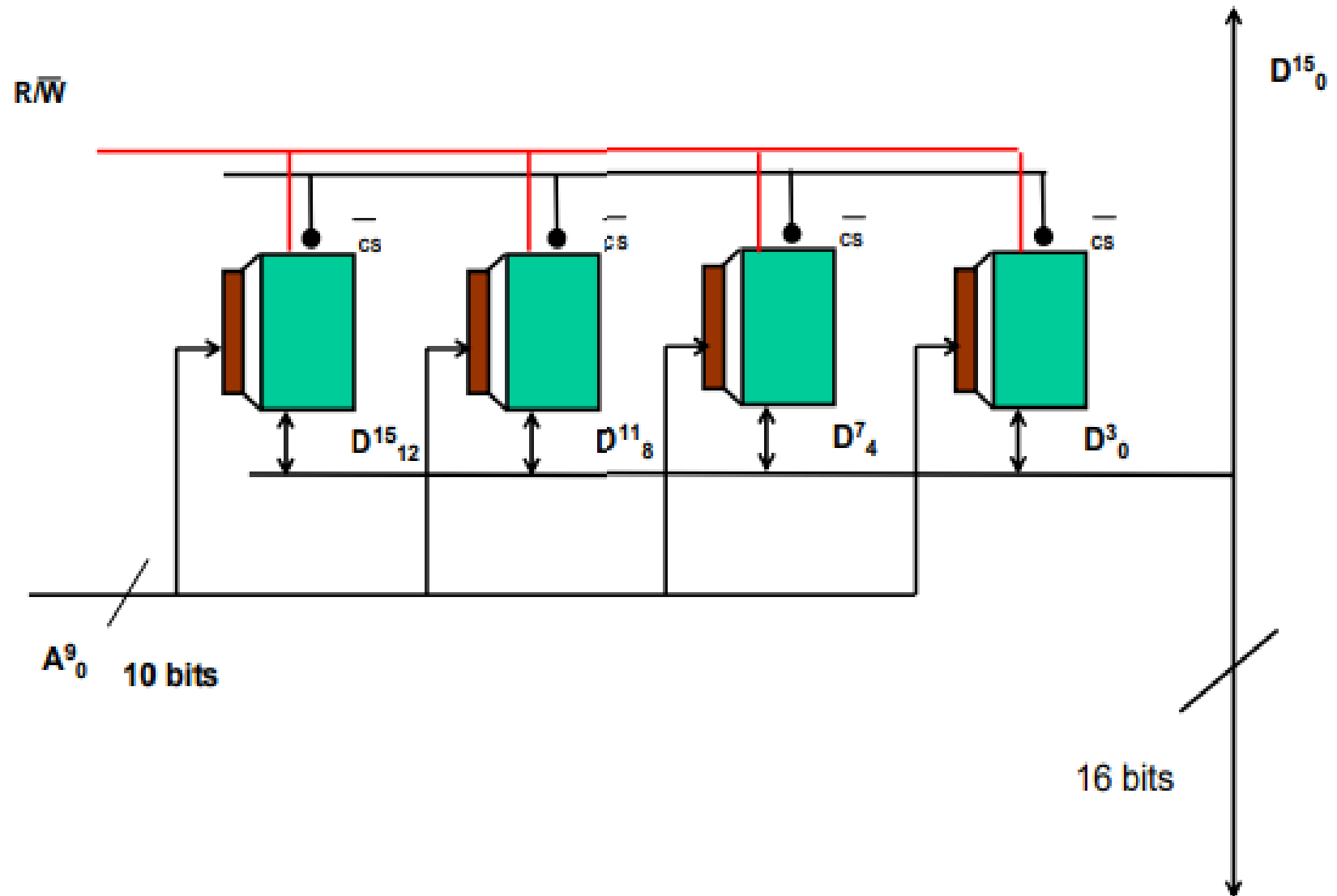
□ Exemple 2

- On veut réaliser une mémoire de 1Ko (la taille d'un mot est de 16 bits) en utilisant des boîtiers de taille 1Ko mots de 4 bits) ?

- **Solution :**

- $(m,n)=(1024,16) \rightarrow$ taille du bus d'adresses est de 10 bits A^9_0 (A9...A0),
taille de bus de données est du 16 bits D^{15}_0 (D15....D0)
- $(m',n')=(1024,4) \rightarrow$ taille du bus d'adresses est de 10 bits (A9'...A0'),
taille de bus de données est du 4 bits (D3'....D0')
- $P = 1024/1024 = 1$ (extension lignes)
- $Q = 16/4 = 4$ (extension colonnes)
- Le nombre totale de boîtiers $P.Q = 4$

□ Exemple 2:



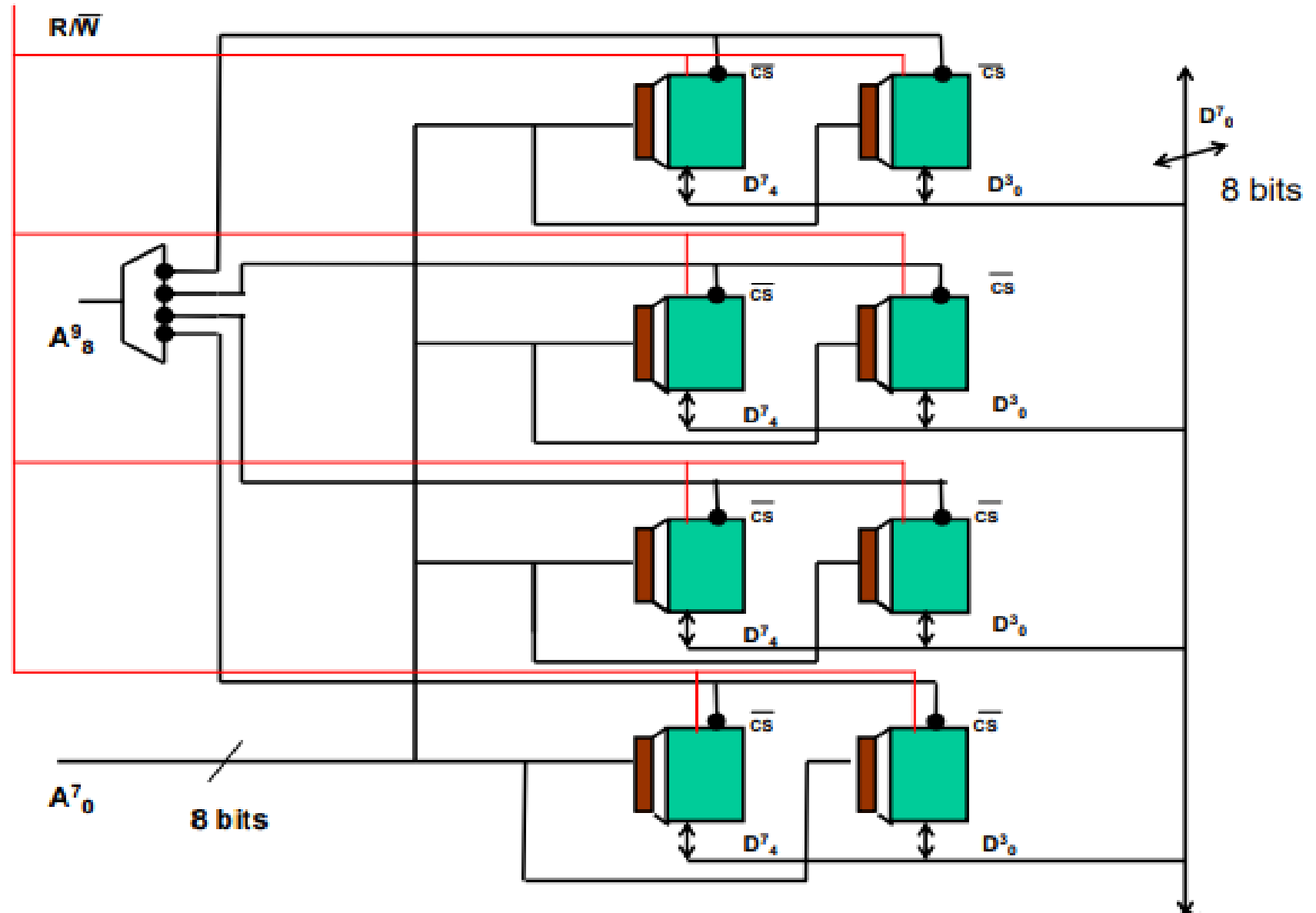
□ Exemple 3 :

- On veut réaliser une mémoire de 1KO (la taille d'un mot est de 8 bits) en utilisant des boîtiers de taille 256 mots de 4 bits) ?

- **Solution :**

- $(m,n)=(1024,8) \rightarrow$ taille du bus d'adresses est de 10 bits ($A9...A0$),
taille du bus de données est de 8 bits ($D7....D0$)
- $(m',n')=(256,4) \rightarrow$ taille du bus d'adresses est de 8 bits ($A7...A0$),
taille du bus de données est de 4 bits ($D3....D0$)
- $P = 1024/256 = 4$ (extension lignes)
- $Q = 8/4 = 2$ (extension colonnes)
- Le nombre totale de boîtiers $P.Q=8$

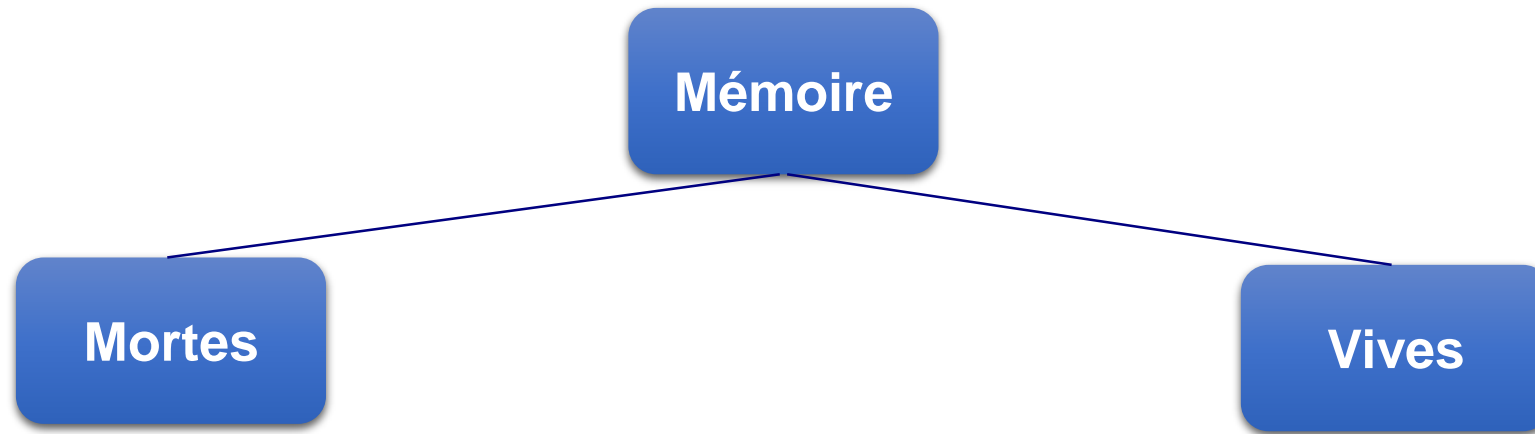
❑ Exemple 3:



□ Exercice

Réaliser une mémoire de 8K X12 (la taille d'un mot est de 12 bits) en utilisant des boîtiers de taille 2048 mots de 4 bits)

❑ Différents types de mémoire

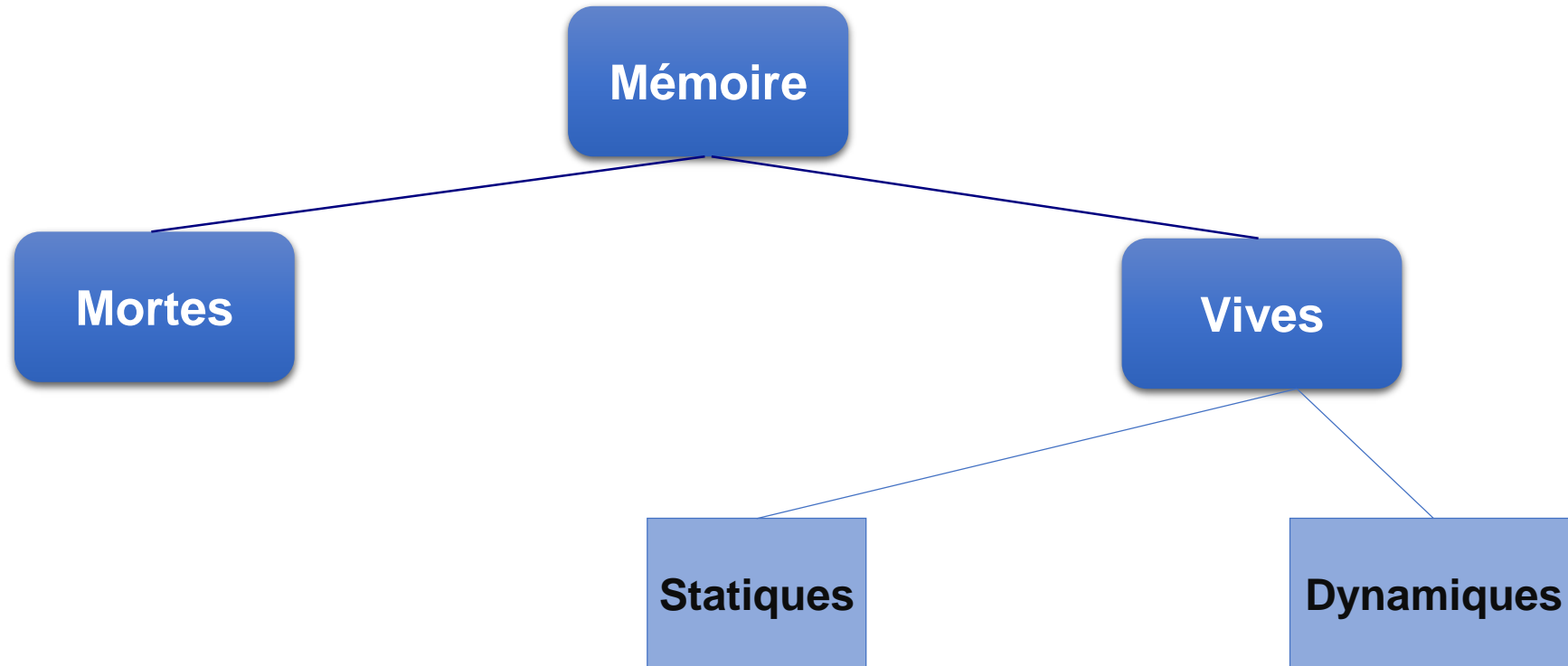


❑ Différents types de mémoire

❖ RAM (Random Access Memory):

- Mémoire vive : stockage temporaire de données
- Lecture et écriture possible
- Le contenu est modifiable
- Mémoire volatile = Perte des informations hors alimentation électrique
- RAM **statique** ou **dynamique**
- Elle doit avoir un temps de cycle très court pour ne pas ralentir le microprocesseur.

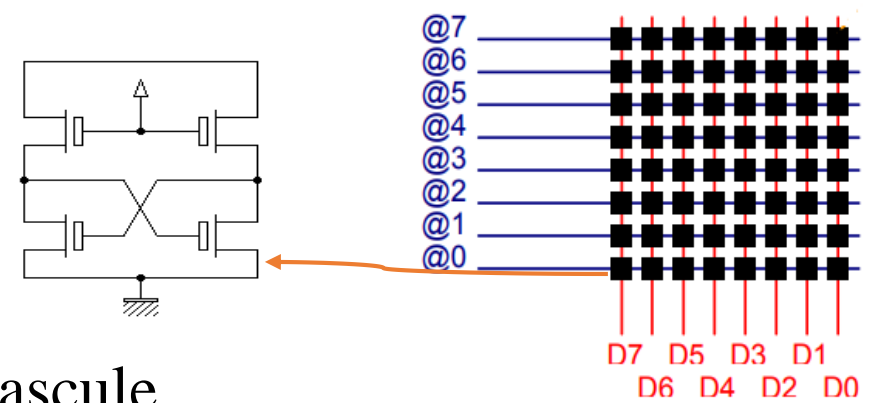
❑ Différents types de mémoire



□ Différents types de mémoire

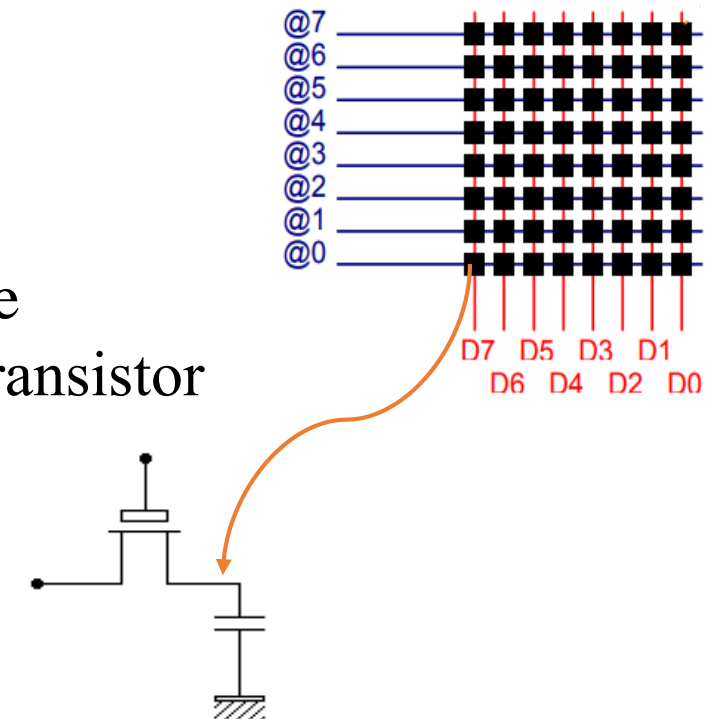
▪ RAM statique (SRAM)

- Le bit mémoire d'une SRAM est composé d'une bascule.
- Une bascule est un circuit logique capable de maintenir les valeurs de ses sorties malgré les changements de valeurs d'entrées, c'est-à-dire comportant un état « mémoire ». La sortie peut être au niveau logique 0 ou 1.
- Chaque bascule contient entre **4 et 6 transistors**.



▪ RAM dynamique (DRAM)

- L'information est mémorisée sous la forme d'une charge électrique stockée dans un condensateur, en plus d'un transistor de commande.
- Le condensateur détermine la valeur du bit
= **1 chargé, 0 non chargé**



❑ Différents types de mémoire

Comparaison:

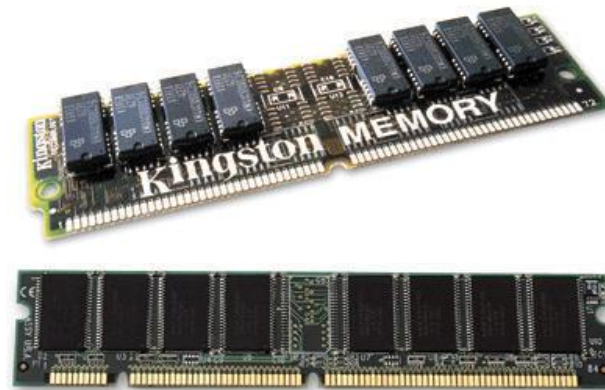
- **DRAM** : - Nécessite un rafraîchissement périodique de l'information
 - L'information est perdue si on ne la régénère pas périodiquement (charge du condensateur).
 - La durée de ces actions augmente le temps d'accès aux informations.
 - Coût: faible
- **SRAM** : - Ne nécessite pas de rafraîchissement
 - : - Beaucoup plus rapide que la DRAM
 - Coût: Coûteuse
 - Consommation d'énergie: Faible

❑ Différents types de mémoire

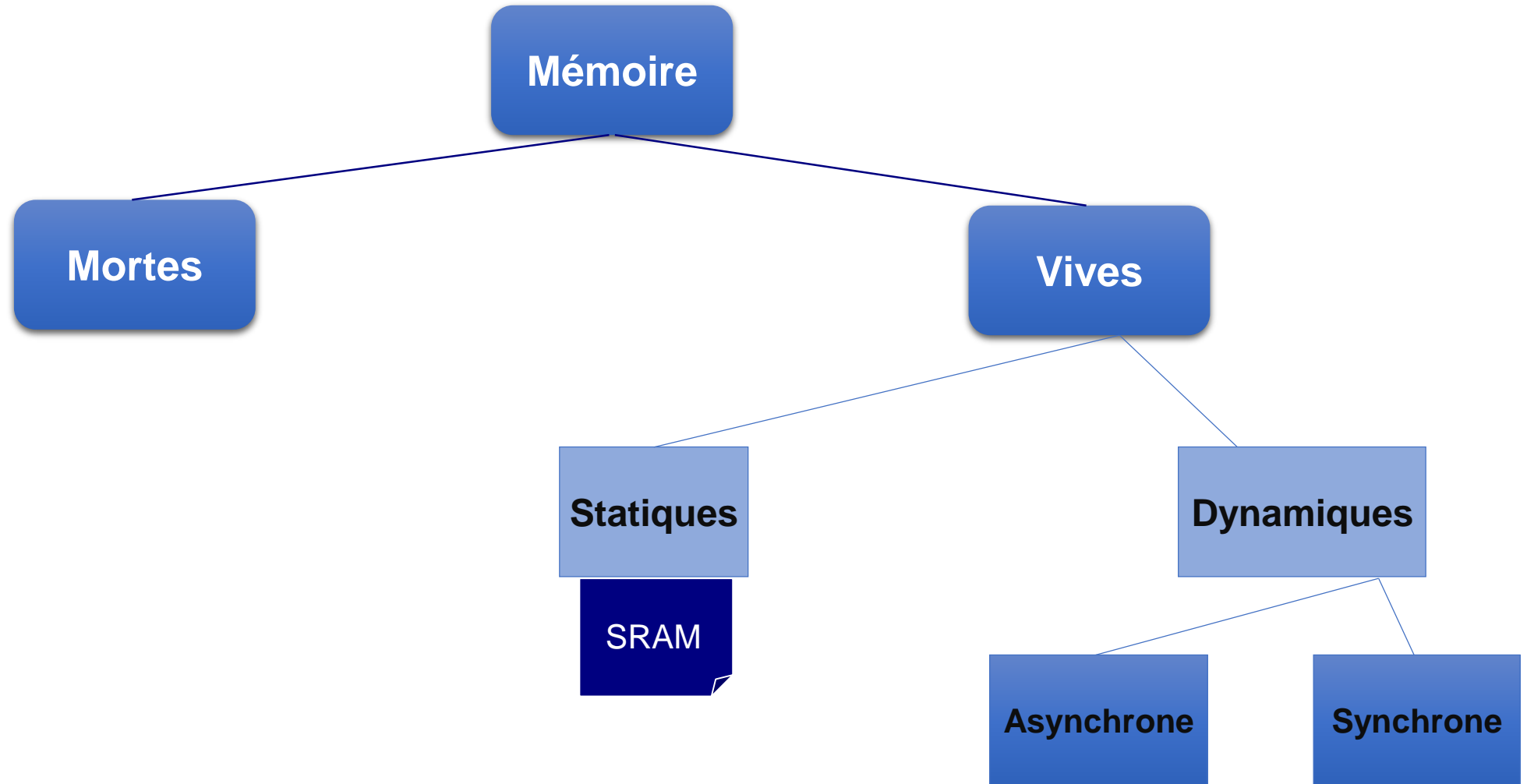
- Application : RAM statique / RAM dynamique

SRAM = mémoire cache de petite taille et rapide

DRAM = mémoire principale de forte capacité et de faible coût



❑ Différents types de mémoire



□ Différents types de mémoire

■ DRAM:

❖ Mémoire asynchrone

- Pour ce type de mémoire, l'intervalle de temps entre deux accès mémoire consécutif n'est pas régulier. Le processeur ne sait donc pas quand l'information qu'il attend est disponible et doit attendre (wait-state) (**temps d'attente**) que la mémoire lui transmette les données.
- Pour être sûr de l'avoir, le processeur n'a comme solution que d'attendre que la mémoire lui transmette les données. S'il ne le faisait pas, il pourrait la manquer et cela provoquerait des erreurs.
- Ces mémoires ont disparu de nos ordinateurs modernes

□ Différents types de mémoire

■ DRAM:

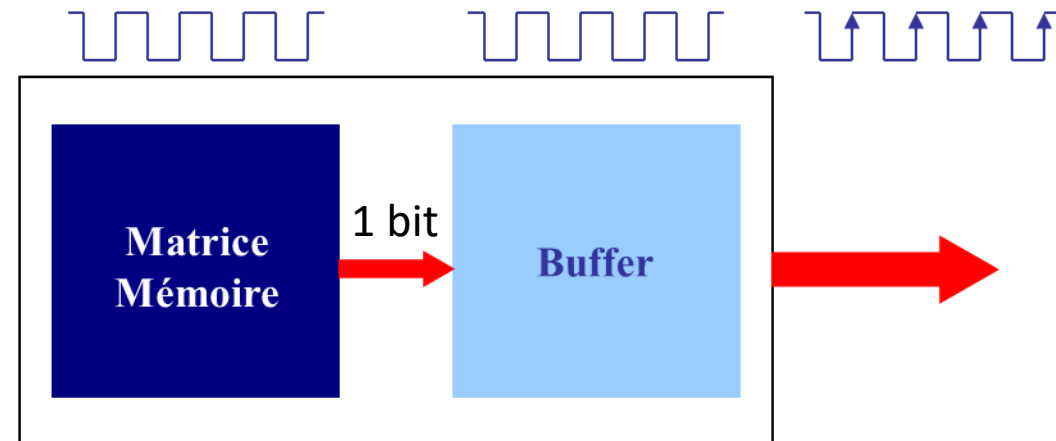
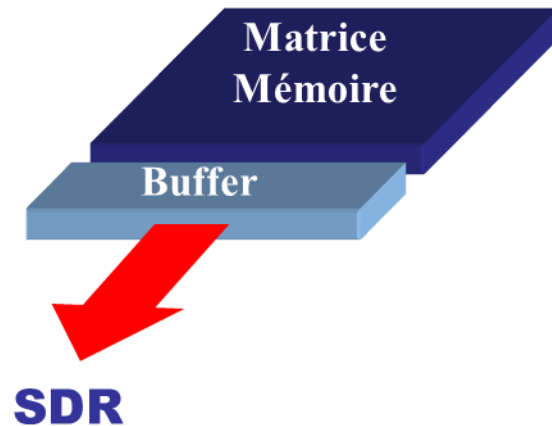
❖ Mémoire synchrone :

- Lorsque le processeur fait un accès à la mémoire, il peut continuer à travailler en attendant la réponse. Par exemple, faire un autre accès mémoire.
- La cadence de sortie des informations est **régulière**, on évite ainsi les états d'attente (wait state) du processeur.
- La mémoire est reliée à une **horloge**. Celle-ci cadence donc les échanges entre la mémoire et le processeur. → **le processeur sait quand l'information sera disponible**
- La mémoire et le processeur sont reliés à un **buffer** qui stocke les demandes d'accès à la mémoire provenant du microprocesseur.

□ Différents types de mémoire

❖ Quelques DRAM

- **Synchronous DRAM (SDRAM)**
 - Une matrice de cellules mémoires
 - Un buffer d'Entrée/Sortie
 - Un bus de données

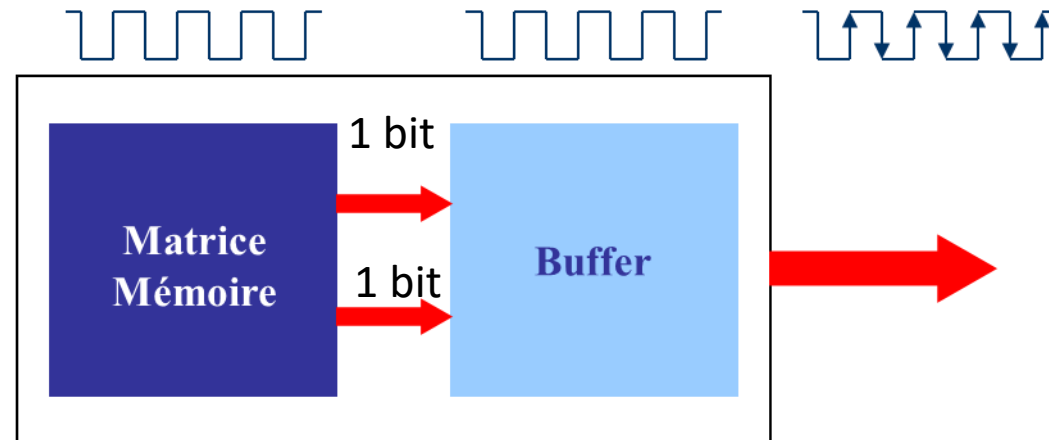
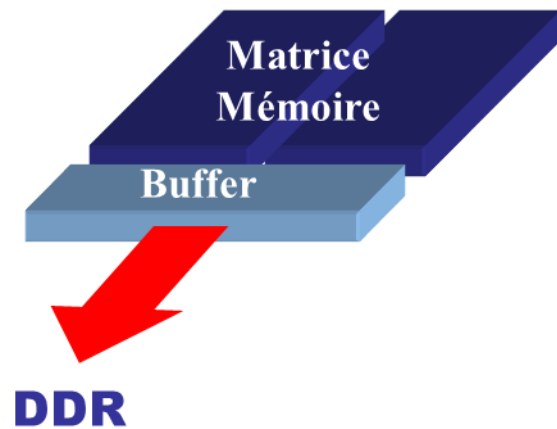


□ Différents types de mémoire

❖ Quelques DRAM

▪ Double Data Rate SDRAM

- basée sur la technologie SDRAM
- Le taux de transfert est doublé
- Lecture sur front montant et descendant
- Banc mémoire X2

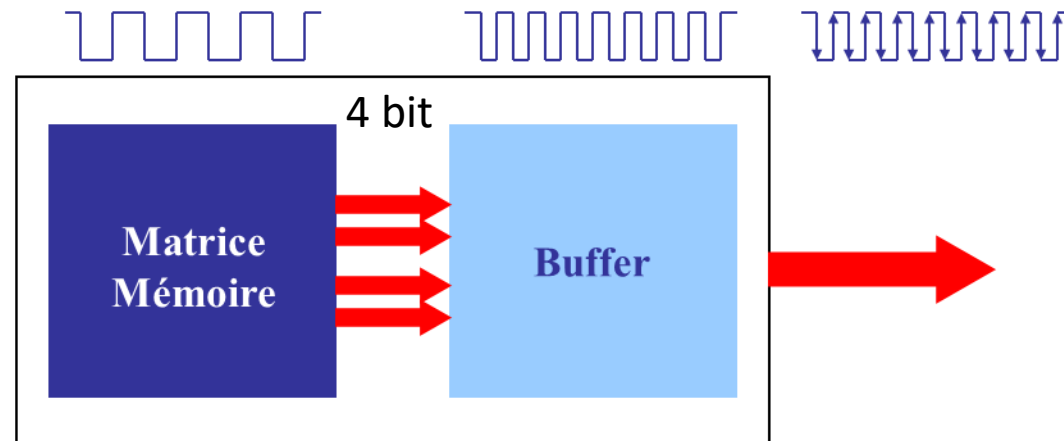
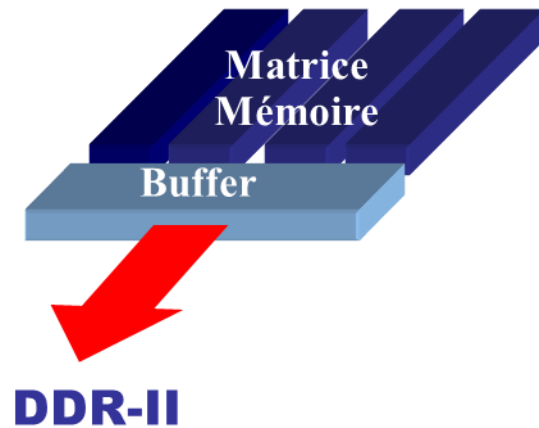


❑ Différents types de mémoire

❖ Quelques DRAM

▪ Double Data Rate II SRAM

- Vitesse du buffer x 2
- Banc mémoire x 2



❑ Différents types de mémoire

❖ Quelques DRAM

▪ **DDR3-SDRAM (Troisième génération)**

- Elle apporte une série d'améliorations par rapport à la génération précédente et offre de meilleures performances.
- Elle diminue la consommation électrique de 40 %
- Le débit théorique de ces barrettes peut dépasser les 10 Go/s.

▪ **DDR4-SDRAM**

Le progrès apporté par la DDR4 par rapport à son prédécesseur concerne :

- Sa vitesse/ fréquence est meilleure
- La capacité s'est également améliorée
- Les barrettes de mémoire DDR4 consomment moins d'énergie

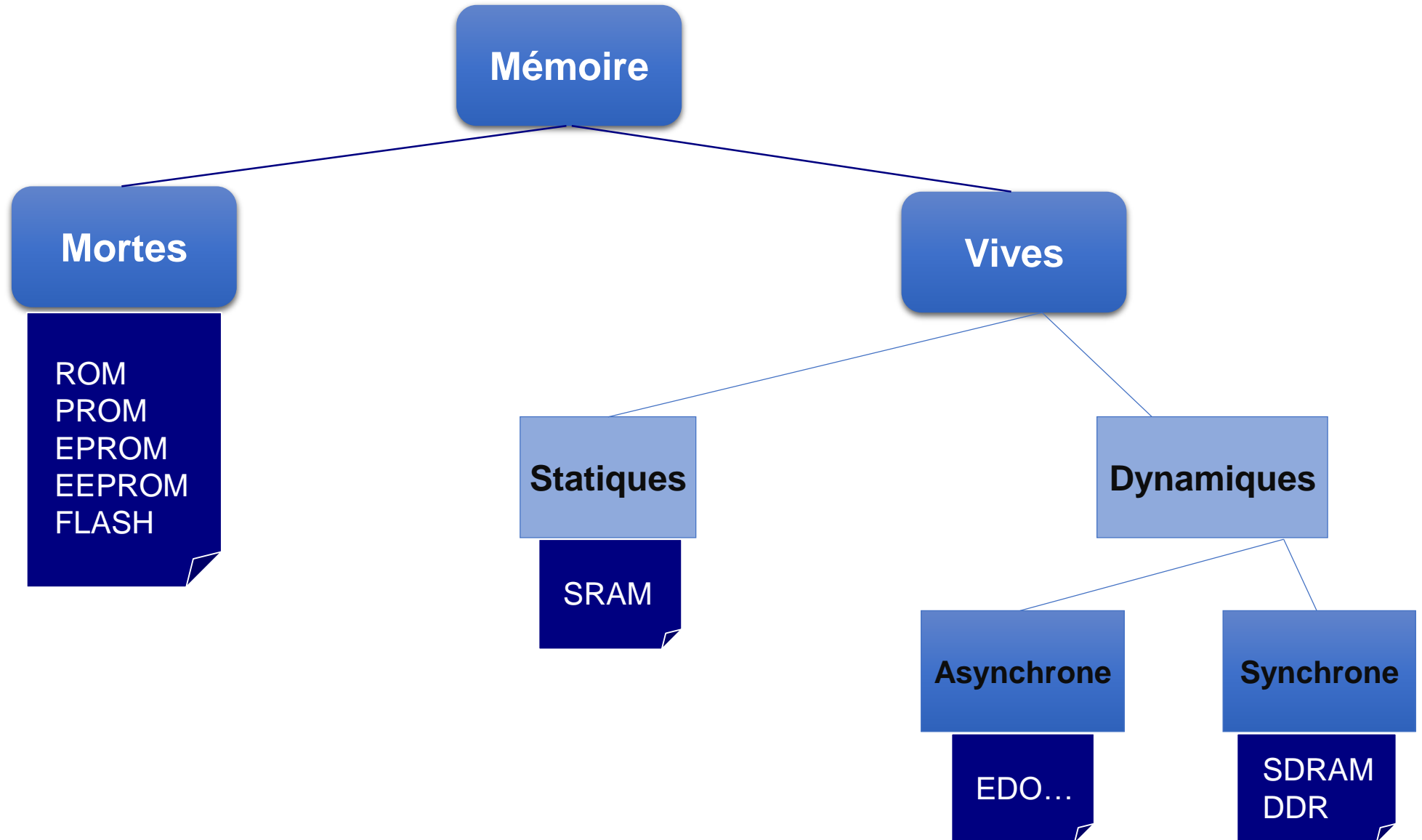
□ Différents types de mémoire

❖ Quelques DRAM

▪ DDR5-SDRAM

- L'apparition de la DRAM DDR5 était en **2020-2021**.
- Elle offre des performances **deux fois plus élevées** que celles de la mémoire DDR4 tout en étant **plus économe en énergie**.

❑ Différents types de mémoire



❑ Différents types de mémoire

❖ ROM (Read Only Memory)

- Mémoire morte
- Accessible seulement en Lecture
- Mémoire non volatile
- Mémoire rapide
- Écriture effectuée par l'intermédiaire d'un programmeur spécifique



❑ Différents types de mémoire

❖ Types de ROM:

- **PROM ((Programmable ROM)**
 - Mémoire programmable une seule fois
 - Programmation par l'utilisateur
 - Modification impossible
 - Coût faible

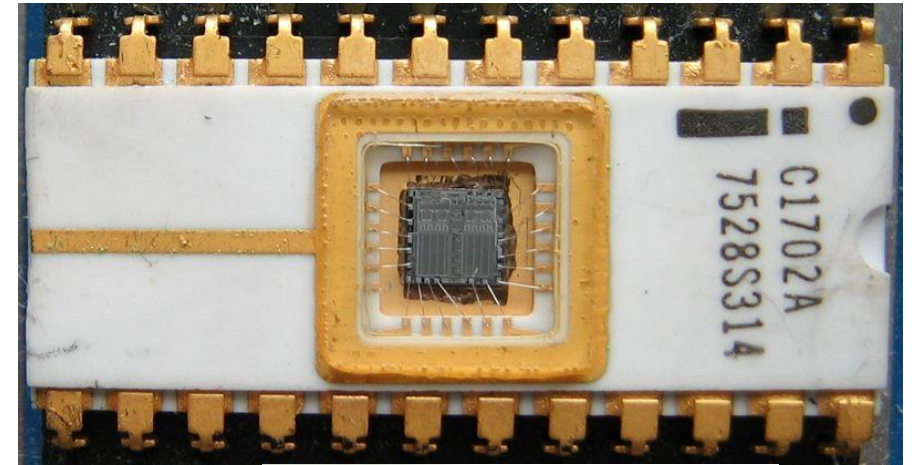


❑ Différents types de mémoire

❖ Types de ROM:

▪ EPROM (Erasable PROM)

- une PROM qui peut être effacée
- Programmation par pic de tension
- Effaçable entièrement par UV
- Reprogrammable et non Volatile
- Programmation hors système
- Effacement hors système
- Temps d'effacement long (15 minutes pour les UV-EPROM)
- l'écriture est beaucoup plus lente que sur une RAM. (environ 1000x)



□ Différents types de mémoire

❖ Types de ROM:

- **EEPROM (Electrically EPROM)**
 - Programmation par pic de tension
 - Effaçable mot par mot par pic de tension
 - Temps d'effacement rapide
 - Programmation et effacement ISP (*In-System Programming*)
 - Comportement d'une RAM non volatile mais très lente

❑ Différents types de mémoire

❖ Types de ROM:

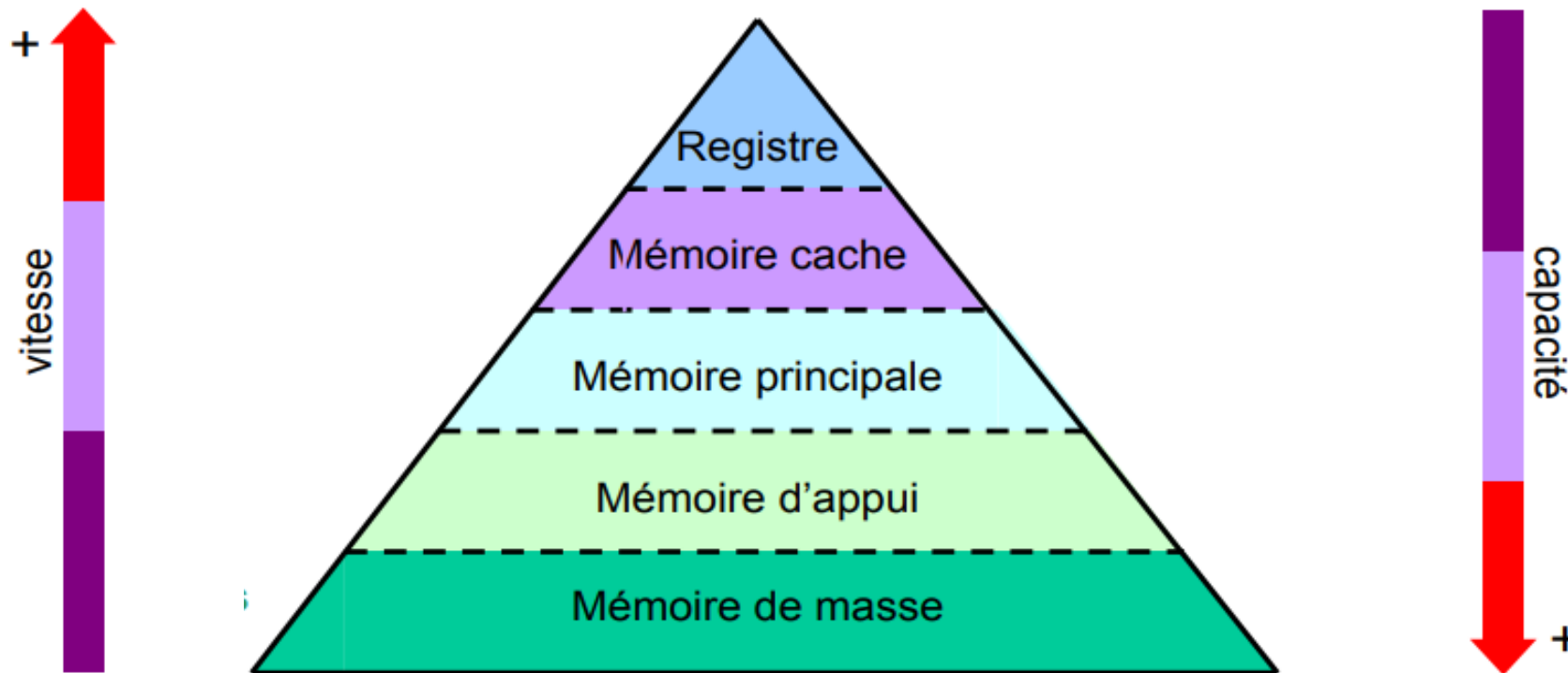
▪ Flash EEPROM

- Programmation et effacement ISP
- Programmation par mot ou bloc
- Temps d'effacement très rapide



□ Hiérarchie mémoire

- Dans un ordinateur, plusieurs niveaux de mémoire



Les mémoires doivent répondre à deux contraintes contradictoires :

- Taille importante
- Temps d'accès court

□ Hiérarchie mémoire

❖ Les registres

- Cellules mémoire situées dans le processeur (CPU)
- Servent à gérer l'exécution des **opérations**:
 - **Spécialisés**, ex : adresse de la prochaine instruction en mémoire
 - **Généraux** : résultats intermédiaires des calculs
- **Taille globale faible**
 - Jusqu'à quelques dizaines/centaines octets
- **Très rapides** (vitesse du CPU)

□ Hiérarchie mémoire

❖ Les registres dans la mémoire centrale

- Il existe 2 types de registres:
 - D'adresse (contient l'adresse d'un mot) RA
 - De données = registres « mot » (contiennent un mot mémoire) RM
- Tailles des registres :
 - Registre mot RM : taille d'un **mot mémoire**
 - Registre d'adresses RA : en général, taille d'un **mot mémoire**, ou moins
- Le registre d'adresses doit permettre d'adresser **toutes** les cellules de la mémoire → la taille maximale de mémoire que le CPU peut gérer dépend **uniquement** de la taille du registre d'adresses

□ Hiérarchie mémoire

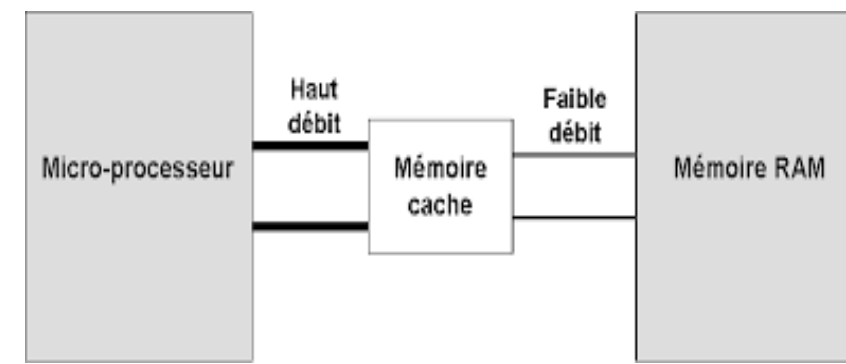
❖ Mémoire cache

■ Problèmes

- En pratique, les accès à la mémoire centrale sont très lents; une trop grande différence de vitesse entre le CPU et la mémoire centrale.
 - ➡ La vitesse du CPU est plus rapide que la MP.
 - ➡ Le CPU gaspille des millions de cycles d'horloge!
 - ➡ Le processeur ne fonctionne pas au meilleur rythme

■ Solution

- Une mémoire qui va stocker les informations (instructions, données) les **plus fréquemment utilisées** par le CPU lors de l'exécution d'un programme.
- On intègre cette mémoire entre le CPU et la MP pour **améliorer la vitesse** de transfert des informations entre ces deux derniers.
 - ➡ Cette mémoire va permettre au CPU de faire **moins d'accès** à la MP et ainsi de **gagner du temps**.



❑ Hiérarchie mémoire

❖ Mémoire cache

- Mémoire intermédiaire entre le processeur et la mémoire centrale
- Mémoire intégrée dans le processeur et cadencée à la même fréquence
- **But de la mémoire cache :**
 - Débit de la mémoire centrale très lent par rapport au débit requis par le processeur
 - ➔ On **accélère la vitesse de lecture** des informations par le CPU en les plaçant (**en avance**) dans le cache
- **Mémoire volatile**
- De type **SRAM** car doit être rapide
- **Faible capacité**, Taille : jusqu'à quelques dizaines de Mo
- Invisible pour le système d'exploitation.

□ Hiérarchie mémoire

❖ Mémoire cache

■ Problèmes

- Mémoire cache doit être petite (quelques centaines de Ko ou quelques Mo) pour être efficace en terme de débit
- Ne peut donc pas y stocker tout un programme et ses données

■ Solutions

- Algorithmes pour « deviner » et mettre dans le cache les données/instructions avant que le CPU en ait besoin
- Recherche bon compromis entre tailles, types de cache (données/instructions), niveaux de cache, techniques d'accès au cache ... pour meilleures performances

□ Hiérarchie mémoire

❖ Mémoire cache

- Les programmes standard ont souvent des comportements prédictibles soit temporellement, soit spatialement, de sorte que **les accès mémoires** réalisés par le processeur pour les exécuter **ne se font pas au hasard**.

=> Les caches utilisent ce principe:

- Sauvegardent les informations les plus **récemment accédées**, en cas de **ré-access**

□ Hiérarchie mémoire

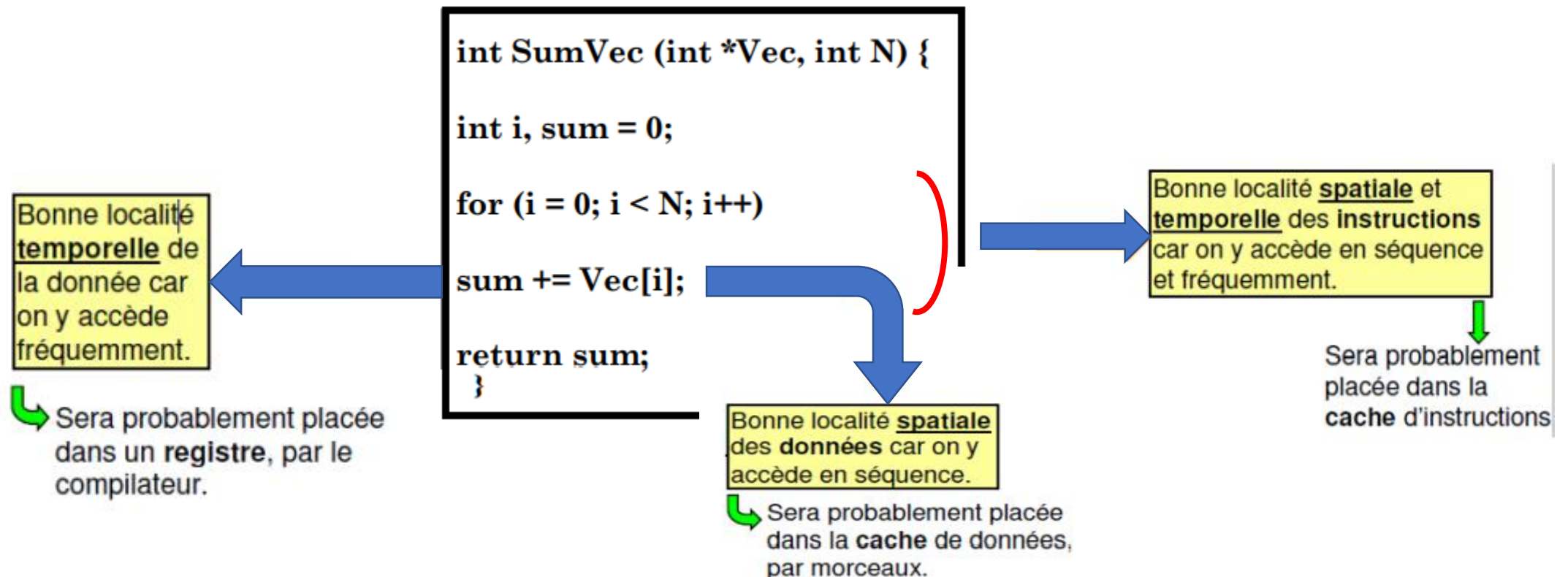
❖ Mémoire cache: Principe de localité

- La mémoire cache est basée sur le principe de localité :
 - **Localité temporelle** : plus un mot mémoire a été accédé récemment, plus il est probable qu'il soit ré-accédé à nouveau (exemple: boucle).
 - cas des boucles de quelques instructions seulement
 - **Localité spatiale** : plus un mot mémoire est proche du dernier mot mémoire accédé, plus il est probable qu'il soit accédé (exemple: tableau).
 - cas des instructions exécutées en séquence

❑ Hiérarchie mémoire

❖ Mémoire cache: Principe de localité

- Exemple: Est-ce que le morceau de code ci-dessous présente de bonnes caractéristiques de localité ?



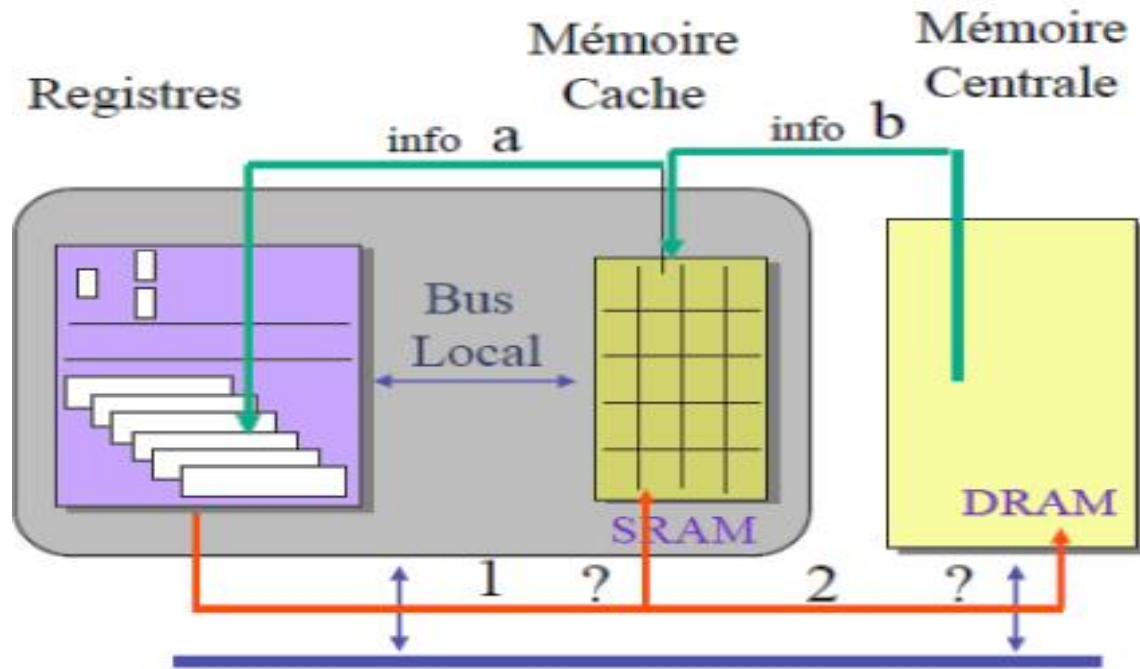
□ Hiérarchie mémoire

❖ Mémoire cache: Fonctionnement

- Le cache contient des copies d'informations (instructions ou données) qui sont en mémoire centrale.
- Avant tout accès à la mémoire, le processeur vérifie si les informations ne sont pas présentes dans le cache.
- Si le processeur utilise les informations contenues dans le cache et n'accède pas à la mémoire centrale. On parle alors de **succès de cache** (cache hit).
- Sinon, on parle de défaut de cache ou **d'échec de cache** (cache miss) dans le cas où il est nécessaire d'aller chercher les données en mémoire centrale.

❑ Hiérarchie mémoire

❖ Mémoire cache: Fonctionnement



1. L'info cherchée est-elle dans le cache ?

OUI / **Succès** (a) : ramener l'info dans le processeur

NON / **Défaut** (2) : chercher l'info dans la mémoire centrale

2. L'info est-elle en mémoire centrale ?

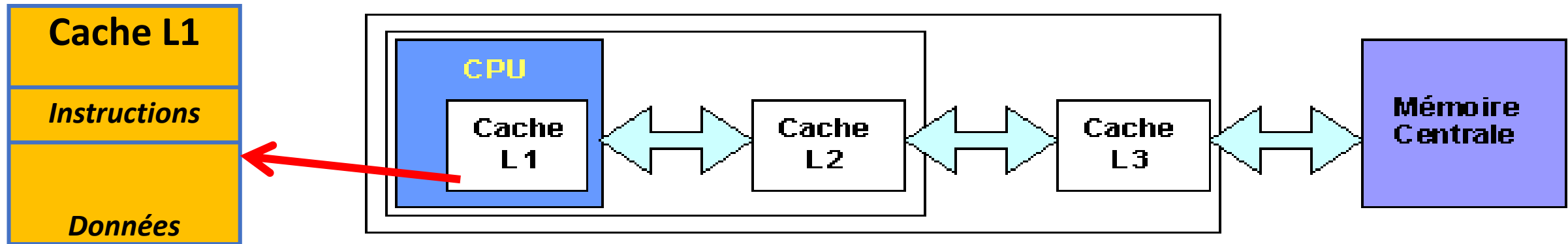
OUI : (b) ramener l'info dans le cache , puis dans le processeur (a)

NON : ramener l'info dans la MC puis dans le cache (b)

- La mémoire cache n'apporte un gain de performance que dans le premier **cas (a)**. Sa performance est donc entièrement liée à son taux de succès.
- Il est courant de rencontrer des taux de succès moyen de l'ordre de 80 à 90%

❑ Hiérarchie mémoire

❖ Mémoire cache: Niveaux de caches



- Cache du niveau L_{i+1} joue le rôle de cache pour le niveau L_i
- Cache L_{i+1} plus grand que L_i mais moins rapide en temps d'accès aux données
- En général, deux niveaux de cache: L1 (**classiquement de 32 Ko à 128 Ko**) et L2 (**typiquement de 512 Ko à 2 Mo**).
- Les caches du niveau L1 sont divisés en 2 parties Instructions/Données
=> (Permet de doubler la bande passante entre le cache et le processeur)

□ Hiérarchie mémoire

❖ Mémoire cache: Niveaux de caches

□ Relation entre les niveaux de cache:

- **Cache inclusif:**

- Le contenu du niveau L1 se trouve aussi dans L2
- Taille globale du cache : celle du cache L2

- **Cache exclusif**

- Le contenu des niveaux L1 et L2 sont différents
- Taille globale du cache : taille L1 + taille L2

□ Hiérarchie mémoire

❖ Mémoire centrale

- Elle contient les programmes (**instructions et données**)
- Taille : jusqu'à plusieurs (dizaines de) Go
- **Accès direct**
- De type **DRAM**
- Vitesse relativement lente

□ Hiérarchie mémoire

❖ La mémoire d'appui

- mémoire **intermédiaire** entre la mémoire centrale et les mémoires de masse
- Elle joue le **même rôle** que la mémoire cache.
- Elle est présente dans les ordinateurs les plus évolués et permet d'augmenter la vitesse d'échange des informations entre ces deux niveaux.

□ Hiérarchie mémoire

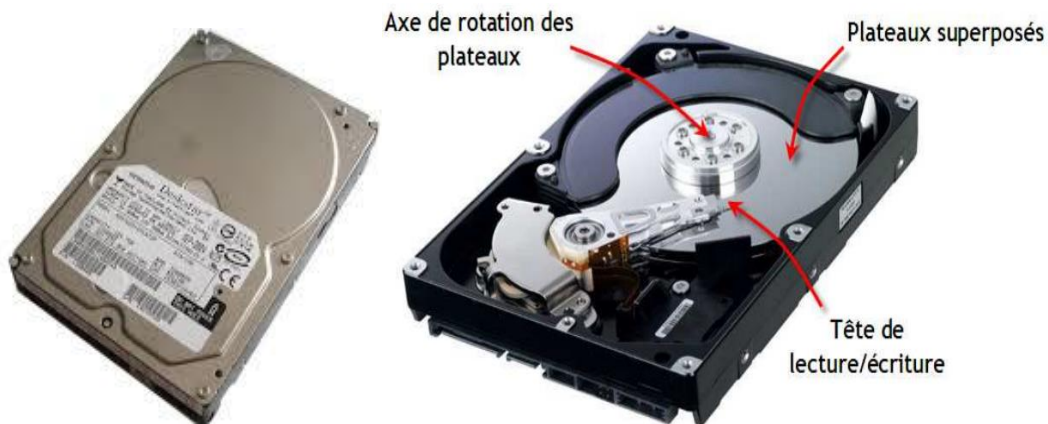
❖ Mémoire de masse

- Mémoire de **grande capacité** : de quelques Go à plusieurs To
- Mémoire **non volatile**: stockage permanent
- Stockage de programmes, données, fichiers en tout genre
- **Très lente**
- Exemples :
 - Disque dur
 - Clés USB
 - DVD

❑ Hiérarchie mémoire

❖ Mémoire de masse: Disque Dur

- Le disque dur est l'un des éléments importants de l'ordinateur servant à conserver les données de manière permanente.
- Différent types:
 - HDD (Hard Drive Disk)
 - SSD (Solid State Drive)



❑ Critères de choix d'une mémoire

- Les principaux critères à retenir sont :
 - **Capacité**
 - **Vitesse**
 - **Consommation**
 - **Coût**